



Sun Java System Directory Server Enterprise Edition 6.2 Migration Guide



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 820-2490
September, 2007

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and SunTM Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certains composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Contents

Preface	13
1 Overview of the Migration Process for Directory Server	23
Before You Migrate	23
Prerequisites to Migrating a Single Directory Server Instance From 5.1 and Later	24
Deciding on the New Product Distribution	24
Outline of Migration Steps	25
Deciding on Automatic or Manual Migration	25
2 Automated Migration Using the <code>dsmig</code> Command	27
About the Automatic Migration Tool	27
Prerequisites for Running <code>dsmig</code>	28
Using <code>dsmig</code> to Migrate the Schema	29
Using <code>dsmig</code> to Migrate Security Data	29
Using <code>dsmig</code> to Migrate Configuration Data	30
Plug-in Configuration Data	30
Chained Suffix Configuration Data	31
Configuration Data For Suffixes With Multiple Backends	31
Replication Configuration Data	31
Configuration Data for <code>o=netscapeRoot</code>	31
Configuration Attributes Not Migrated by <code>dsmig</code>	32
Using <code>dsmig</code> to Migrate User Data	33
Tasks to be Performed After Automatic Migration	34
3 Migrating Directory Server Manually	35
Before You Start a Manual Migration	35
Migrating the Schema Manually	36

Migrating Configuration Data Manually	37
Migration of Specific Configuration Attributes	37
Migrating Security Settings Manually	46
Migrating User Data Manually	47
Migrating User Plug-Ins Manually	48
Tasks to be Performed After Manual Migration	48
4 Migrating a Replicated Topology	49
Overview of Migrating Replicated Servers	49
Issues Related to Migrating Replicated Servers	50
Issues With the New Password Policy	50
Migration of Replication Agreements	50
Migration of Referrals	50
Manual Reset of Replication Credentials	51
Problems Related to Tombstone Purging	51
New Replication Recommendations	51
Migration Scenarios	52
Migrating a Replicated Topology to an Identical Topology	52
Migrating a Replicated Topology to a New Topology	61
Migrating Over Multiple Data Centers	65
5 Architectural Changes in Directory Server	67
Changes in the Administration Framework	67
Removal of the <i>ServerRoot</i> Directory	67
Removal of the <i>o=netscapeRoot</i> Suffix	68
Changes to ACIs	68
Changes in the ACI Scope	68
Changes in Suffix-Level ACIs	68
Command Line Changes	69
Deprecated Commands	71
Changes to the Console	72
New Password Policy	72
Password Policy Compatibility	73
Changes to Plug-Ins	78
New Plug-Ins in Directory Server 6.2	78

Plug-Ins Deprecated in Directory Server 6.2	78
Changes to the Plug-In API	79
Changes to the Installed Product Layout	79
Administration Utilities Previously Under <i>ServerRoot</i>	79
Binaries Previously Under <i>ServerRoot/bin</i>	80
Libraries and Plug-Ins Previously Under <i>ServerRoot/lib</i>	80
Online Help Previously Under <i>ServerRoot/manual</i>	80
Plug-Ins Previously Under <i>ServerRoot/plugins</i>	80
Utilities Previously Under <i>ServerRoot/shared/bin</i>	81
Certificate and Key Files	82
Silent Installation and Uninstallation Templates	82
Server Instance Scripts Previously Under <i>ServerRoot/slapd-ServerID</i>	83
Server Instance Subdirectories	83
6 Migrating Directory Proxy Server	85
Mapping the Global Configuration	85
Mapping the Global Security Configuration	87
Mapping the Connection Pool Configuration	89
Mapping the Groups Configuration	90
Mapping the Group Object	90
Mapping the Network Group Object	91
Mapping Bind Forwarding	92
Mapping Operation Forwarding	93
Mapping Subtree Hiding	94
Mapping Search Request Controls	94
Mapping Compare Request Controls	95
Mapping Attributes Modifying Search Requests	95
Mapping Attributes Restricting Search Responses	96
Mapping the Referral Configuration Attributes	97
Mapping the Server Load Configuration	98
Mapping the Properties Configuration	99
Attribute Renaming Property	99
Forbidden Entry Property	99
LDAP Server Property	100
Load Balancing Property	101

Search Size Limit Property	103
Log Property	103
Mapping the Events Configuration	105
Mapping the Actions Configuration	106
Configuring Directory Proxy Server 6.2 as a Simple Connection-Based Router	106
7 Migrating Identity Synchronization for Windows	107
Migration Overview	108
Before You Migrate Identity Synchronization for Windows	108
Preparing for Identity Synchronization for Windows Migration	109
Exporting Version 1.1 Configuration	109
Checking for Undelivered Messages	116
▼ Using the checktopics Utility	116
▼ To Clear Messages	117
Forcing Password Changes on Windows NT	118
Migrating Your System	118
Preparing for Migration	119
▼ Preparing to migrate from version 1.1, and 1.1 SP1, to version 6.0	120
Uninstalling Identity Synchronization for Windows	122
▼ To Uninstall Identity Synchronization for Windows Version 1.1	122
Installing or Upgrading the Dependent Products	124
Installing Identity Synchronization for Windows 6.0	124
▼ To install the Identity Synchronization for Windows 6.0 components:	124
What to Do if the 1.1 Uninstallation Fails	127
Manually Uninstalling 1.1 Core and Instances from Solaris	127
▼ To Manually Uninstall Core From a Solaris Machine:	128
Manually Uninstalling 1.1 Core and Instances from Windows 2000	132
▼ To uninstall Core from a Windows 2000 machine:	133
▼ Manually Uninstalling a 1.1 Instance from Windows NT	137
Other Migration Scenarios	141
Multi-Master Replication Deployment	142
Multi-Host Deployment with Windows NT	143
Checking the Logs	146
Index	147

Figures

FIGURE 4-1	Existing version 5 Topology	53
FIGURE 4-2	Isolating the Consumer From the Topology	53
FIGURE 4-3	Migrating the version 5 Consumer	54
FIGURE 4-4	Placing the 6.0 Consumer Into the Topology	55
FIGURE 4-5	Existing version 5 Topology With Migrated Consumers	56
FIGURE 4-6	Isolating the Hub From the Topology	56
FIGURE 4-7	Migrating the version 5 Hub	57
FIGURE 4-8	Placing the 6.0 Hub Into the Topology	58
FIGURE 4-9	Existing version 5 Topology With Consumers and Hubs Migrated	59
FIGURE 4-10	Isolating the Master From the Topology	60
FIGURE 4-11	Migrating the version 5 Master	60
FIGURE 4-12	Placing the 6.0 Master Into the Topology	61
FIGURE 4-13	Existing version 5 Topology	62
FIGURE 4-14	Existing Topology With Migrated Servers	63
FIGURE 4-15	Migrated Topology With Promoted Hub Replicas	64
FIGURE 4-16	New Fully-Meshed All-Master Topology	65
FIGURE 7-1	Migrating a Single-Host Deployment	119
FIGURE 7-2	Migrating a Multi-Master Replication Deployment	143
FIGURE 7-3	Migrating a Multi-Host Deployment with Windows NT	145

Tables

TABLE 1-1	Migration Matrix Showing Support for Automated Migration	26
TABLE 3-1	Change Log Attribute Name Changes	40
TABLE 3-2	Fractional Replication Attribute Name Changes	40
TABLE 3-3	Mapping Between 5 and 6.2 Password Policy Attributes	42
TABLE 5-1	Directory Server 5 and 6 commands	69
TABLE 5-2	Directory Server 5 and 6 Commands (Subcommands of the <code>directoryserver</code> Command)	71
TABLE 5-3	Version 5 Commands That Have Been Deprecated	71
TABLE 5-4	Directory Server Password Policy Mode Interoperability	77
TABLE 5-5	Support for Plug-Ins	81
TABLE 5-6	Tools Previously Under <i>ServerRoot</i> /shared/bin	81
TABLE 5-7	Location of Certificate and Key Files	82
TABLE 5-8	Instance-Specific Subdirectories	83
TABLE 6-1	Mapping of Version 5 Global Configuration Attributes to 6.0 Properties	86
TABLE 6-2	Mapping of Security Configuration	88
TABLE 6-3	Mapping of Connection Pool Attributes	89
TABLE 6-4	Mapping Between Version 5 Group Attributes and Version 6 Connection Handler Properties	90
TABLE 6-5	Mapping Between Version 5 Network Group Attributes and 6.2 Properties	91
TABLE 6-6	Mapping of Directory Proxy Server 5 Bind Forwarding Attributes to Directory Proxy Server 6 Connection Handler Property Settings	92
TABLE 6-7	Mapping of Directory Proxy Server 5 Operation Forwarding Attributes to Directory Proxy Server 6 Request Filtering Properties	93
TABLE 6-8	Mapping Directory Proxy Server 5 Search Request Control Attributes to Directory Proxy Server 6.2 Properties	95
TABLE 6-9	Mapping of Directory Proxy Server 5 Compare Request Control Attributes to Directory Proxy Server 6 Properties	95
TABLE 6-10	Mapping of Directory Proxy Server 5 Search Request Modifying Attributes to Directory Proxy Server 6 Properties	96
TABLE 6-11	Mapping of Directory Proxy Server 5 Search Response Restriction Attributes to	

	Directory Proxy Server 6.2 Properties	97
TABLE 6-12	Mapping of Directory Proxy Server 5 Referral Configuration Attributes to Directory Proxy Server 6 resource limits Properties	98
TABLE 6-13	Mapping of Directory Proxy Server 5 Server Load Configuration Attributes to Directory Proxy Server 6.2 Resource Limits Properties	98
TABLE 6-14	Mapping of Directory Proxy Server 5 Server Load Configuration Attributes to Directory Proxy Server 6 Resource Limits Properties	100
TABLE 6-15	Mapping of ids-proxy-sch-LDAPServer Attributes to Data Source Properties	101
TABLE 6-16	Mapping of Version 5 Search Size Limit Attributes to 6.0 Properties	103
TABLE 6-17	Version 5 and Version 6 Log Functionality	104
TABLE 6-18	Mapping Between Version 5 Event Attributes and Version 6 Connection Handler Properties	105
TABLE 7-1	Component Distribution in a Multi-Master Replication Deployment	142
TABLE 7-2	Multi-Host Deployment	144

Examples

EXAMPLE 7-1 Sample Export Configuration File 111

Preface

This *Migration Guide* describes how to migrate the components of Directory Server Enterprise Edition to version 6.2. The guide provides migration instructions for Directory Server, Directory Proxy Server, and Identity Synchronization for Windows.

Who Should Use This Book

This guide is intended for directory service administrators who are migrating to Directory Server Enterprise Edition 6.2. The guide might also be useful to business planners who are considering migrating to the new version.

Before You Read This Book

If you are not yet familiar with this version of Directory Server Enterprise Edition, you might want to start by evaluating the new features and capabilities of the product. For more information, see the *Sun Java System Directory Server Enterprise Edition 6.0 Evaluation Guide* and the *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes*.

How This Book Is Organized

[Chapter 1, “Overview of the Migration Process for Directory Server”](#) describes the steps involved in migrating to Directory Server 6.2.

[Chapter 2, “Automated Migration Using the `dsmig` Command”](#) explains how to use the migration tool provided with Directory Server 6.2.

[Chapter 3, “Migrating Directory Server Manually”](#) describes the process for manual migration of each part of Directory Server.

[Chapter 4, “Migrating a Replicated Topology”](#) describes the issues involved in migrating replicated servers.

[Chapter 5, “Architectural Changes in Directory Server”](#) describes the architectural changes in Directory Server 6.2 that affect migration from a previous version.

Chapter 6, “Migrating Directory Proxy Server” describes how the configuration properties in Directory Proxy Server 6.2 can be used to simulate a version 5 configuration.

Chapter 7, “Migrating Identity Synchronization for Windows” describes the steps involved in migrating to Identity Synchronization for Windows 6.2.

Directory Server Enterprise Edition Documentation Set

This Directory Server Enterprise Edition documentation set explains how to use Sun Java System Directory Server Enterprise Edition to evaluate, design, deploy, and administer directory services. In addition, it shows how to develop client applications for Directory Server Enterprise Edition. The Directory Server Enterprise Edition documentation set is available at <http://docs.sun.com/coll/1224.3>.

For an introduction to Directory Server Enterprise Edition, review the following documents in the order in which they are listed.

TABLE P-1 Directory Server Enterprise Edition Documentation

Document Title	Contents
<i>Sun Java System Directory Server Enterprise Edition 6.2 Release Notes</i>	Contains the latest information about Directory Server Enterprise Edition, including known problems.
<i>Sun Java System Directory Server Enterprise Edition 6.2 Documentation Center</i>	Contains links to key areas of the documentation set.
<i>Sun Java System Directory Server Enterprise Edition 6.2 Evaluation Guide</i>	Introduces the key features of this release. Demonstrates how these features work and what they offer in the context of a fictional deployment that you can implement on a single system.
<i>Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide</i>	Explains how to plan and design highly available, highly scalable directory services based on Directory Server Enterprise Edition. Presents the basic concepts and principles of deployment planning and design. Discusses the solution life cycle, and provides high-level examples and strategies to use when planning solutions based on Directory Server Enterprise Edition.
<i>Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide</i>	<p>Explains how to install the Directory Server Enterprise Edition software. Shows how to select which components to install, configure those components after installation, and verify that the configured components function properly.</p> <p>For instructions on installing Directory Editor, go to http://docs.sun.com/coll/DirEdit_05q1.</p> <p>Make sure you read the information in <i>Sun Java System Directory Server Enterprise Edition 6.2 Release Notes</i> related to Directory Editor before you install Directory Editor.</p>

TABLE P-1 Directory Server Enterprise Edition Documentation (Continued)

Document Title	Contents
<i>Sun Java System Directory Server Enterprise Edition 6.2 Migration Guide</i>	Provides instructions for upgrading components from earlier versions of Directory Server, Directory Proxy Server, and Identity Synchronization for Windows.
<i>Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide</i>	<p>Provides command-line instructions for administering Directory Server Enterprise Edition.</p> <p>For hints and instructions on using the Directory Service Control Center (DSCC) to administer Directory Server Enterprise Edition, see the online help provided in DSCC.</p> <p>For instructions about administering Directory Editor, go to http://docs.sun.com/coll/DirEdit_05q1.</p> <p>For instructions about installing and configuring Identity Synchronization for Windows, see Part II, “Installing Identity Synchronization for Windows,” in <i>Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide</i>.</p>
<i>Sun Java System Directory Server Enterprise Edition 6.2 Developer’s Guide</i>	Provides instructions for developing directory client applications with the tools and APIs that are provided as part of Directory Server Enterprise Edition.
<i>Sun Java System Directory Server Enterprise Edition 6.2 Reference</i>	Introduces the technical and conceptual foundations of Directory Server Enterprise Edition. Describes its components, architecture, processes, and features. Also provides a reference to the developer APIs.
<i>Sun Java System Directory Server Enterprise Edition 6.2 Man Page Reference</i>	Describes the command-line tools, schema objects, and other public interfaces that are available through Directory Server Enterprise Edition. Individual sections of this document can be installed as online manual pages.
<i>Sun Java System Directory Server Enterprise Edition 6.2 Troubleshooting Guide</i>	Provides information for defining the scope of the problem, gathering data, and troubleshooting the problem areas using various tools.
<i>Sun Java System Identity Synchronization for Windows 6.0 Deployment Planning Guide</i>	Provides general guidelines and best practices for planning and deploying Identity Synchronization for Windows.

Related Reading

The SLAMD Distributed Load Generation Engine is a Java™ application that is designed to stress test and analyze the performance of network-based applications. It was originally developed by Sun Microsystems, Inc. to benchmark and analyze the performance of LDAP directory servers. SLAMD is available as an open source application under the Sun Public License, an OSI-approved open source license. To obtain information about SLAMD, go to <http://www.slamd.com/>. SLAMD is also available as a java.net project. See <https://slamd.dev.java.net/>.

Java Naming and Directory Interface (JNDI) technology supports accessing the Directory Server using LDAP and DSML v2 from Java applications. For information about JNDI, see <http://java.sun.com/products/jndi/>. The *JNDI Tutorial* contains detailed descriptions and

examples of how to use JNDI. This tutorial is at <http://java.sun.com/products/jndi/tutorial/>.

Directory Server Enterprise Edition can be licensed as a standalone product, as a component of Sun Java Enterprise System, as part of a suite of Sun products, such as the Sun Java Identity Management Suite, or as an add-on package to other software products from Sun. Java Enterprise System is a software infrastructure that supports enterprise applications distributed across a network or Internet environment. If Directory Server Enterprise Edition was licensed as a component of Java Enterprise System, you should be familiar with the system documentation at <http://docs.sun.com/coll/1286.3>.

Identity Synchronization for Windows uses Message Queue with a restricted license. Message Queue documentation is available at <http://docs.sun.com/coll/1307.2>.

Identity Synchronization for Windows works with Microsoft Windows password policies.

- Information about password policies for Windows 2003 is available in the [Microsoft documentation](#) online.
- Information about changing passwords, and about group policies in Windows 2003 is available in the [Microsoft documentation](#) online.
- Information about the Microsoft Certificate Services Enterprise Root certificate authority is available in the [Microsoft support documentation](#) online.
- Information about configuring LDAP over SSL on Microsoft systems is available in the [Microsoft support documentation](#) online.

Redistributable Files

Directory Server Enterprise Edition does not provide any files that you can redistribute.

Default Paths and Command Locations

This section explains the default paths used in the documentation, and gives the locations of commands on different operating systems and deployment types.

Default Paths

The table in this section describes the default paths that are used in this document. For complete descriptions of the files installed, see the following product documentation.

- Chapter 14, “Directory Server File Reference,” in *Sun Java System Directory Server Enterprise Edition 6.2 Reference*
- Chapter 25, “Directory Proxy Server File Reference,” in *Sun Java System Directory Server Enterprise Edition 6.2 Reference*
- Appendix A, “Directory Server Resource Kit File Reference,” in *Sun Java System Directory Server Enterprise Edition 6.2 Reference*

TABLE P-2 Default Paths

Placeholder	Description	Default Value
<i>install-path</i>	Represents the base installation directory for Directory Server Enterprise Edition software. The software is installed in directories below this base <i>install-path</i> . For example, Directory Server software is installed in <i>install-path/ds6/</i> .	When you install from a zip distribution using <code>dsee_deploy(1M)</code> , the default <i>install-path</i> is the current directory. You can set the <i>install-path</i> using the <code>-i</code> option of the <code>dsee_deploy</code> command. When you install from a native package distribution, such as you would using the Java Enterprise System installer, the default <i>install-path</i> is one of the following locations: <ul style="list-style-type: none"> ■ Solaris systems - <code>/opt/SUNWdsee/</code>. ■ Red Hat systems - <code>/opt/sun/</code>. ■ Windows systems - <code>C:\Program Files\Sun\JavaES5\DSEE</code>.
<i>instance-path</i>	Represents the full path to an instance of Directory Server or Directory Proxy Server. The documentation uses <code>/local/ds/</code> for Directory Server and <code>/local/dps/</code> for Directory Proxy Server.	No default path exists. Instance paths must nevertheless always be found on a <i>local</i> file system. The following directories are recommended: <code>/var</code> on Solaris systems <code>/global</code> if you are using Sun Cluster
<i>serverroot</i>	Represents the parent directory of the Identity Synchronization for Windows installation location	Depends on your installation. Note the concept of a <i>serverroot</i> no longer exists for Directory Server.
<i>isw-hostname</i>	Represents the Identity Synchronization for Windows instance directory	Depends on your installation
<i>/path/to/cert8.db</i>	Represents the default path and file name of the client’s certificate database for Identity Synchronization for Windows	<i>current-working-dir/cert8.db</i>

TABLE P-2 Default Paths (Continued)

Placeholder	Description	Default Value
<i>serverroot/ismw-hostname/logs/</i>	Represents the default path to the Identity Synchronization for Windows local logs for the System Manager, each connector, and the Central Logger	Depends on your installation
<i>serverroot/ismw-hostname/logs/central/</i>	Represents the default path to the Identity Synchronization for Windows central logs	Depends on your installation

Command Locations

The table in this section provides locations for commands that are used in Directory Server Enterprise Edition documentation. To learn more about each of the commands, see the relevant man pages.

TABLE P-3 Command Locations

Command	Java ES, Native Package Distribution	Zip Distribution
cacaoadm	Solaris - <i>/usr/sbin/cacaoadm</i>	Solaris - <i>install-path/dsee6/cacao_2/usr/sbin/cacaoadm</i>
	Red Hat - <i>/opt/sun/cacao/bin/cacaoadm</i>	Red Hat- <i>install-path/dsee6/cacao_2/cacao/bin/cacaoadm</i>
	Windows - <i>install-path\share\cacao_2\bin\cacoadm.bat</i>	Windows - <i>install-path\dsee6\cacao_2\bin\cacoadm.bat</i>
certutil	Solaris - <i>/usr/sfw/bin/certutil</i>	<i>install-path/dsee6/bin/certutil</i>
	Red Hat- <i>/opt/sun/private/bin/certutil</i>	
dpadm(1M)	<i>install-path/dps6/bin/dpadm</i>	<i>install-path/dps6/bin/dpadm</i>
dpconf(1M)	<i>install-path/dps6/bin/dpconf</i>	<i>install-path/dps6/bin/dpconf</i>
dsadm(1M)	<i>install-path/ds6/bin/dsadm</i>	<i>install-path/ds6/bin/dsadm</i>

TABLE P-3 Command Locations (Continued)

Command	Java ES, Native Package Distribution	Zip Distribution
dsccon(1M)	<i>install-path/dscc6/bin/dsccon</i>	<i>install-path/dscc6/bin/dsccon</i>
dsccreg(1M)	<i>install-path/dscc6/bin/dsccreg</i>	<i>install-path/dscc6/bin/dsccreg</i>
dscsetup(1M)	<i>install-path/dscc6/bin/dscsetup</i>	<i>install-path/dscc6/bin/dscsetup</i>
dsconf(1M)	<i>install-path/ds6/bin/dsconf</i>	<i>install-path/ds6/bin/dsconf</i>
dsee_deploy(1M)	Not provided	<i>install-path/dsee6/bin/dsee_deploy</i>
dsmig(1M)	<i>install-path/ds6/bin/dsmig</i>	<i>install-path/ds6/bin/dsmig</i>
entrycmp(1)	<i>install-path/ds6/bin/entrycmp</i>	<i>install-path/ds6/bin/entrycmp</i>
fildif(1)	<i>install-path/ds6/bin/fildif</i>	<i>install-path/ds6/bin/fildif</i>
idsktune(1M)	Not provided	At the root of the unzipped zip distribution
insync(1)	<i>install-path/ds6/bin/insync</i>	<i>install-path/ds6/bin/insync</i>
ns-accountstatus(1M)	<i>install-path/ds6/bin/ns-accountstatus</i>	<i>install-path/ds6/bin/ns-accountstatus</i>
ns-activate(1M)	<i>install-path/ds6/bin/ns-activate</i>	<i>install-path/ds6/bin/ns-activate</i>
ns-inactivate(1M)	<i>install-path/ds6/bin/ns-inactivate</i>	<i>install-path/ds6/bin/ns-inactivate</i>
repldisc(1)	<i>install-path/ds6/bin/repldisc</i>	<i>install-path/ds6/bin/repldisc</i>
schema_push(1M)	<i>install-path/ds6/bin/schema_push</i>	<i>install-path/ds6/bin/schema_push</i>
smcwebserver	Solaris and Linux - <i>/usr/sbin/smcwebserver</i>	This command pertains only to DSCC when it is installed using native packages distribution.
	Windows - <i>install-path\share\webconsole\bin\smcwebserver</i>	
wcadmin	Solaris and Linux - <i>/usr/sbin/wcadmin</i>	This command pertains only to DSCC when it is installed using native packages distribution.
	Windows - <i>install-path\share\webconsole\bin\wcadmin</i>	

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P-4 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name% su</code> Password:
<i>AaBbCc123</i>	A placeholder to be replaced with a real name or value	The command to remove a file is <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized (note that some emphasized items appear bold online)	Read Chapter 6 in the <i>User's Guide</i> . A <i>cache</i> is a copy that is stored locally. Do <i>not</i> save the file.

Shell Prompts in Command Examples

The following table shows default system prompts and superuser prompts.

TABLE P-5 Shell Prompts

Shell	Prompt
C shell on UNIX and Linux systems	<code>machine_name%</code>
C shell superuser on UNIX and Linux systems	<code>machine_name#</code>
Bourne shell and Korn shell on UNIX and Linux systems	<code>\$</code>
Bourne shell and Korn shell superuser on UNIX and Linux systems	<code>#</code>
Microsoft Windows command line	<code>C:\</code>

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P-6 Symbol Conventions

Symbol	Description	Example	Meaning
[]	Contains optional arguments and command options.	<code>ls [-l]</code>	The <code>-l</code> option is not required.
{ }	Contains a set of choices for a required command option.	<code>-d {y n}</code>	The <code>-d</code> option requires that you use either the <code>y</code> argument or the <code>n</code> argument.
<code>\${ }</code>	Indicates a variable reference.	<code>\${com.sun.javaRoot}</code>	References the value of the <code>com.sun.javaRoot</code> variable.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (<http://www.sun.com/documentation/>)
- Support (<http://www.sun.com/support/>)
- Training (<http://www.sun.com/training/>)

Searching Sun Product Documentation

Besides searching Sun product documentation from the docs.sun.comSM web site, you can use a search engine by typing the following syntax in the search field:

```
search-term site:docs.sun.com
```

For example, to search for “broker,” type the following:

```
broker site:docs.sun.com
```

To include other Sun web sites in your search (for example, java.sun.com, www.sun.com, and developers.sun.com), use `sun . com` in place of `docs . sun . com` in the search field.

Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to <http://docs.sun.com> and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 820-2490.

Overview of the Migration Process for Directory Server

This chapter describes the steps involved in migrating to Directory Server 6.2. Directory Server 6.2 provides a migration tool, `dsmig`, that automates aspects of the migration for certain platform/version combinations. If servers within your topology fall outside of these combinations, the same migration steps must be performed manually.

This chapter includes the following topics:

- “Before You Migrate” on page 23
- “Deciding on the New Product Distribution” on page 24
- “Outline of Migration Steps” on page 25
- “Deciding on Automatic or Manual Migration” on page 25

Before You Migrate

This chapter provides an overview of the upgrade and data migration process.

Before upgrading, familiarize yourself with the new features and fixes available in the current version. Take the opportunity to review design decisions made during implementation of existing directory services. For a description of all new features and fixes, see “What’s New at a Glance” in *Sun Java System Directory Server Enterprise Edition 6.2 Evaluation Guide*. For information about the new features that specifically affect migration, see [Chapter 5](#), “Architectural Changes in Directory Server.”

Prerequisites to Migrating a Single Directory Server Instance From 5.1 and Later

Before migrating from a 5.1 or later server instance, ensure that the following prerequisites are met:

- Directory Server 6.2 must be installed. The new server can be installed on the same machine as the existing server or on a different machine.
- Ensure that the new machine has sufficient local disk space to house binaries and databases for both the old and new servers, and also enough extra space to hold LDIF files containing the entries in all existing suffixes. You can estimate the local disk space required as somewhat larger than the following calculation.

$$\text{local space required} = 2 * (\text{space for existing server}) + (\text{space for LDIF files})$$

- If you are using the automatic migration tool, the following two prerequisites must be met:
 - The existing server instance must be stopped cleanly.
 - If the new server is located on a different machine, a complete image of the original server instance must be created on the new machine. This includes all schema files, configuration files, security files, and database files, in an identical layout to the original server root.

To determine whether you should use automatic or manual migration, see [“Deciding on Automatic or Manual Migration” on page 25](#).

- If your Directory Server deployment includes Identity Synchronization for Windows, you must uninstall Identity Synchronization for Windows before migrating to Directory Server 6.2. For information about migrating Identity Synchronization for Windows, see [Chapter 7, “Migrating Identity Synchronization for Windows.”](#)

Deciding on the New Product Distribution

Directory Server 6.2 is provided in two distributions:

- Java Enterprise System distribution. This distribution takes the form of operating system-specific packages, such as pkg for Solaris and rpm for Linux.
- Compressed archive (zip) distribution.

There are two major differences between these two distributions:

1. Installation from zip can be done anywhere on the system and as a non-root user. The Java Enterprise System distribution requires installation as a super user. It is also more difficult from an automated deployment perspective to install the packages anywhere but in the default location.
2. The zip distribution can be installed as many times as required and multiple distinct versions of the same product can coexist on a single operating system instance. This is not true for the Java Enterprise System distribution. The new version of certain shared component packages required by Directory Server are incompatible with the previous version of these packages. When you migrate to the new version of Directory Server using the Java Enterprise System distribution, the old Directory Server version will no longer run on that machine.

Depending on your environment and the specific requirements of your organization, select the appropriate packaging format.

Outline of Migration Steps

Migration to Directory Server 6.2 can be broken down into the following distinct steps:

1. Migrating the Schema
2. Migrating the Security Settings
3. Migrating the Configuration
4. Migrating the Data
5. Migrating the Plug-Ins
6. Post-migration tasks

To avoid unforeseen problems with the migration, these steps should be performed in the order listed above. In certain cases, you can automate some or all of these steps, using the `dsmig` command. The following section indicates what can be automated and what must be done manually, depending on your existing deployment.

Deciding on Automatic or Manual Migration

This section provides a table that shows when you can use `dsmig` and when you need to migrate manually. It is based on the migration steps described in the previous section.

TABLE 1-1 Migration Matrix Showing Support for Automated Migration

Migrating To		Migration Step				
Software (32/64-bit)	OS	Schema	Config	Security	Data	Plug-Ins
Any	Any	Manual	Manual	Manual	Manual	Manual
Different	Any	dsmig	dsmig	dsmig	Manual	Manual
Same	Different	dsmig	dsmig	dsmig	Manual	Manual
Same	Same	dsmig	dsmig	dsmig	Manual for 5.1 dsmig for 5.2	Manual

The following two chapters explain how to perform each migration step outlined above, either automatically, or manually. For information on automatic migration, see [Chapter 2, “Automated Migration Using the dsmig Command.”](#) For information on manual migration, see [Chapter 3, “Migrating Directory Server Manually.”](#)

Automated Migration Using the `dsmig` Command

Directory Server 6.2 provides a command-line migration tool to help you migrate from a Directory Server 5.1 or 5.2 instance to a Directory Server 6.2 instance. You can only use the migration tool if your deployment satisfies the requirements for automatic migration described in [“Deciding on Automatic or Manual Migration” on page 25](#).

The migration tool provides migration *per instance*. If several instances exist within the same server root, the migration tool must be run for each individual instance.

This chapter explains how to use the migration tool and covers the following topics:

- [“About the Automatic Migration Tool” on page 27](#)
- [“Prerequisites for Running `dsmig`” on page 28](#)
- [“Using `dsmig` to Migrate the Schema” on page 29](#)
- [“Using `dsmig` to Migrate Security Data” on page 29](#)
- [“Using `dsmig` to Migrate Configuration Data” on page 30](#)
- [“Using `dsmig` to Migrate User Data” on page 33](#)
- [“Tasks to be Performed After Automatic Migration” on page 34](#)

About the Automatic Migration Tool

The migration tool, `dsmig`, is delivered with the Directory Server 6.2 packages. When these packages have been installed, `dsmig` is located in `install-path/ds6/bin`.

`dsmig` must be run on the machine on which the new Directory Server instance will be located. When the command is run, a *migration* directory is created within the new instance directory (`new-instance-path/migration`). This directory is a repository for data produced by the migration, including log files and migration status files.

`dsmig` includes a set of sub-commands and options, that map to the individual migration steps described in [“Outline of Migration Steps” on page 25](#). For information about the usage of `dsmig`, see `dsmig(1M)`.

Prerequisites for Running dsmig

In this section, *old instance* refers to the 5.1 or 5.2 instance and *new instance* refers to the Directory Server 6.2 instance.

Before you use dsmig to migrate an instance, ensure that the following tasks have been performed:

- The Directory Server 6.2 packages (either zip, or native packages) have been installed.
The Directory Server 6.2 packages can be installed on the same machine that holds the Directory Server 5.1 instance, or on a different machine.

- The old instance must have been stopped correctly.

A disorderly shutdown of the old instance will cause problems during the migration. Even if the old and new instance are on different machines, the old instance must be stopped before the migration is started.

- dsmig has access to the old instance files.
- If the old and new instances are on different machines, a *complete image* of the old instance must be created on the machine that hosts the new instance.

The complete image includes all the files required for migration of the instance (schema, configuration, security and database files). The complete image files must be located in the same directories as they were under the original Server Root. You can run `cp -r` to achieve this, provided none of the files have been relocated outside the Server Root.

You can create and start the new instance manually, but is not mandatory to create the new instance before running dsmig. dsmig checks whether a new Directory Server instance exists in the specified path. If a new instance exists, the commands are carried out on this instance. If a new instance does exist, the instance is created automatically.

The new instance can be created anywhere except for the exact location of the old instance.

While creating a new instance, a DN and a password for the directory manager is stored in `nsslapd-rootdn` and `nsslapd-rootpw` attributes under `cn=config`. During the migration process, the values for these attributes from the 5.1 or 5.2 instance are not propagated as these attributes already hold a value for the new instance. The same behavior is applied to `nsslapd-secureport` and `nsslapd-port` attributes for the same reason.

Using dsmig to Migrate the Schema

Directory Server 5.1 or 5.2 schema files are located in *serverRoot/slapd-instance-path/config/schema*. Directory Server 6.2 schema files are located in *INSTANCE-PATH/config/schema*.

Directory Server 6.2 provides a new schema file, *00ds6pwp.ldif*, that contains new password policy attributes. In addition, certain configuration attributes have been added to *00core.ldif*.

To migrate the schema automatically, run the following command:

```
$ dsmig migrate-schema old-instance-path new-instance-path
```

When you run this command, any custom schema defined in the *99user.ldif* file are copied to the new instance. If the new instance is already in production, and you have already modified the *99user.ldif* file of the new instance, dsmig performs a *best effort* merge of the two files. Custom schema defined in any other files are also copied to the new instance.

For more information, see *dsmig(1M)*.

Using dsmig to Migrate Security Data

To migrate the security settings automatically, run the following command:

```
$ dsmig migrate-security old-instance-path new-instance-path
```

During the migration of security settings, dsmig performs the following tasks:

- Backs up the certificate and database files in the new instance.
- Copies the certificate database and key database files from the old instance to the new instance.
- Copies the password file from the old instance to the new instance.
- Copies the certificate mapping file from the old instance to the new instance.
- Copies the security module database.

For more information, see *dsmig(1M)*.

Using dsmig to Migrate Configuration Data

Directory Server 5.1 or 5.2 configuration is specified in the file *serverRoot/slapd-instance-path/config/dse.ldif*. Directory Server 6.2 configuration is specified in the file *instance-path/config/dse.ldif*.

To migrate the configuration automatically, run the following command:

```
$ dsmig migrate-config old-instance-path new-instance-path
```

In this step, dsmig reads each LDIF entry in the configuration file (*dse.ldif*) of the 5.1 or 5.2 instance. If these entries exist in the corresponding Directory Server 6.2 configuration file, their values are updated.

Migration of the configuration is done over LDAP. By default, dsmig binds to the new instance securely, issuing a StartTLS request.

Note – By default, StartTLS is not enabled on Windows. If you are running dsmig on Windows, use the `-e` or `--unsecured` option to specify an unsecured connection. Alternatively, use the `-Z` or `--use-secure-port` option to specify a secure connection over SSL. If you do not use either of these options on Windows, dsmig issues a warning and the migration process terminates with an error.

For more information see dsmig(1M). For details of the specific configuration attributes that are migrated, see [“Migration of Specific Configuration Attributes” on page 37](#).

Plug-in Configuration Data

dsmig migrates configuration data for certain Directory Server plug-ins only. For most system plug-ins, configuration data is *not* migrated automatically.

dsmig migrates the following system plug-ins:

- CoS
- 7-bit Check
- DSML Frontend
- Pass-Through Authentication
- Referential Integrity
- Retro Change Log
- UID Uniqueness

When you migrate the configuration in verbose mode, dsmig issues a warning indicating which system plug-in configurations are not migrated.

Plug-ins that you have created are not migrated. However, during the migration process user plug-in configuration data is dumped in the file `new-instance-path/migration/old_userplugins_conf.ldif`. These plug-ins must be recompiled when the migration is complete.

Chained Suffix Configuration Data

Configuration data for chained suffixes is not migrated. By default, the configuration data is dumped in the file `new-instance-path/migration/old_chaining_conf.ldif`. You should not import the `old_chaining_conf.ldif` file in the new instance but use it as a guideline to create the configuration data manually.

Configuration Data For Suffixes With Multiple Backends

Configuration data for suffixes with multiple backends is not migrated. If dsmig detects that a suffix has more than one backend, it does not migrate any of the configuration entries that belong to that suffix. This includes configuration entries for the mapping tree, replicas, replication agreements, LDBM instances, indexes, and encrypted attributes. Instead, all of these entries are dumped in the file `new-instance-path/migration/old_distribution_conf.ldif`.

The entries in the `old_distribution_conf.ldif` file refer to the old instance so should not be imported directly to the new instance. For more information about distribution, see Chapter 22, “Directory Proxy Server Distribution,” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

Replication Configuration Data

Configuration data for replication is not migrated by default. If you want this data to be migrated, use dsmig with the `-R` option. By default, the data is dumped in the file `new-instance-path/migration/old_replication_conf.ldif`. You can import the replication configuration data from this file after migration, if required.

Configuration Data for o=netscapeRoot

Configuration data for the `o=NetscapeRoot` suffix is not migrated by default. If this information is required, use the `-N` option to migrate the configuration data. If you do not use the `-N` option, the data is dumped in the file `new-instance-path/migration/old_netscape_conf.ldif`. You can import the configuration data from this file after migration, if required.

Configuration Attributes Not Migrated by dsmig

The following common configuration attributes are not migrated automatically.

This is not an exhaustive list. You might have used additional configuration attributes that must be migrated manually.

```
ds-hdsml-dsmlschemalocation
ds-hdsml-soapschemalocation
dsKeyedPassword
dsMappedDN
dsMatching-pattern
dsMatching-regexp
dsSaslPluginsEnable
dsSaslPluginsEnable
dsSaslPluginsPath
dsSearchBasedDN
dsSearchFilter
nsabandonedsearchcheckinterval
nsbindconnectionslimit
nsbindretrylimit
nsbindtimeout
nschecklocalaci
nsconcurrentbindlimit
nsconcurrentoperationslimit
nsconnectionlife
nshoplimit
nsMatchingRule
nsmaxresponsedelay
nsmaxtestresponsedelay
nsoperationconnectionslimit
nspossiblechainingcomponents
nspossiblechainingcomponents
nspossiblechainingcomponents
nspossiblechainingcomponents
nspossiblechainingcomponents
nspossiblechainingcomponents
nsproxiedauthorization
nsreferralonscopedsearch
nsslapped-db-durable-transaction
nsslapped-db-home-directory
nsslapped-db-replication-batch-val
nsslapped-db-transaction-logging
nsslapped-directory
nsslapped-disk-full-threshold
nsslapped-disk-low-threshold
nsslapped-exclude-from-export
nsslapped-localhost
```



```
nsslapd-localuser  
nsslapd-mode  
nsslapd-port  
nsslapd-rewrite-rfc1274  
nsslapd-secureport  
nsslapd-security  
nsSSL2  
nsSSL3  
nsSSLActivation  
nsSSLServerAuth  
nsSSLSessionTimeout  
nsState  
nstransmittedcontrols  
plugin-order-preoperation-finish-entry-encode-result
```

Using dsmig to Migrate User Data

In Directory Server 5.2, data is stored in *serverRoot/slapd-instance-name/db*. Directory Server 6.2 stores user data in *instance-path/db*.

To migrate data automatically, run the following command:

```
$ dsmig migrate-data old-instance-path new-instance-path
```

All suffixes are migrated by default, except the *o=netscapeRoot* suffix. dsmig copies the data, the indexes, and the transaction logs. The database context, that is, the state of the database, is not migrated.

In the new Directory Server administration model, there is no Configuration Directory Server. This means that the *o=netscapeRoot* suffix is no longer relevant, unless your deployment includes Identity Synchronization for Windows. By default, dsmig does not migrate the *o=netscapeRoot* database, unless specifically requested. To migrate the *o=netscapeRoot* database, use the *-N* option with the *migrate-data* subcommand.

For more information, see dsmig(1M).

Note – During data migration, Directory Server checks whether nested group definitions exceed 30 levels. Deep nesting can signify a circular group definition, where a nested group contains a group that is also its parent. When a group with more than 30 nesting levels is encountered, Directory Server stops calculating the `isMemberOf` attributes for additional levels.

Each time this happens, Directory Server logs an error. You safely ignore these errors, although you should examine the definition of the group mentioned in the error message for potential circular definitions.

Tasks to be Performed After Automatic Migration

If you have used `dsmig` to migrate your server automatically, only the following two post-migration tasks must be completed:

- If you have customized user plug-ins, these need to be recompiled and added to the new server manually.
- If the migrated server was part of a replicated topology, see [“Issues Related to Migrating Replicated Servers” on page 50](#).

Migrating Directory Server Manually

If your deployment does not satisfy the requirements for automatic migration described in [“Deciding on Automatic or Manual Migration” on page 25](#), you must migrate the servers manually. This chapter describes the process for manual migration of each part of the server.

The chapter covers the following topics:

- [“Before You Start a Manual Migration” on page 35](#)
- [“Migrating the Schema Manually” on page 36](#)
- [“Migrating Configuration Data Manually” on page 37](#)
- [“Migrating Security Settings Manually” on page 46](#)
- [“Migrating User Data Manually” on page 47](#)
- [“Migrating User Plug-Ins Manually” on page 48](#)
- [“Tasks to be Performed After Manual Migration” on page 48](#)

Before You Start a Manual Migration

Migrating an instance manually involves migrating each part of the server in the same order as performed by the automatic migration tool (`dsmig`). In this section, *old instance* refers to the version 5 instance and *new instance* refers to the 6.2 instance.

Before you start a manual migration, ensure that the following tasks have been performed:

- Directory Server 6.2 software has been installed.

Directory Server 6.2 software can be installed on the same machine that holds the Directory Server 5 instance, or on a different machine.

- The new instance has been created.

The new instance can be created anywhere except for the exact location of the old instance. The new instance can be installed on the same LDAP/LDAPS port or on a different port. If you use different ports, any replication agreements to the new instance must be changed accordingly.

While creating a new instance, a DN and a password for the directory manager is stored in `nsslapd-rootdn` and `nsslapd-rootpw` attributes under `cn=config`. During the migration process, the values for these attributes from the 5.2 instance are not propagated as these attributes already hold a value for the new instance. The same behavior is applied to `nsslapd-secureport` and `nsslapd-port` attributes for the same reason.

- The old instance has been stopped correctly.

A disorderly shutdown of the old instance will cause problems during migration. Even if the old and new instances are on different machines, the old instance must be stopped before migration is started.

Migrating the Schema Manually

Directory Server 5 schema files are located in `serverRoot/slapd-serverID/config/schema`.

Directory Server 6.2 schema files are located in `instance-path/config/schema`.

Directory Server 6.2 provides a new schema file, `00ds6pwp.ldif`, that contains new password policy attributes. In addition, certain configuration attributes have been added to `00core.ldif`. Apart from these files, the standard schema files provided with Directory Server 6.2 are identical to those provided in version 5.

To migrate the schema, perform the following steps:

1. Copy the `99user.ldif` file from the existing instance to the new instance. If you have already added custom schema to the new instance, you will need to choose which version of the custom schema to keep.
2. If you have defined custom schema in any other files, copy these files to the new instance.
3. Any fractional replication information must be redefined in the new instance.

Migrating Configuration Data Manually

Directory Server 5 configuration is specified in the file `serverRoot/slapd-serverID/config/dse.ldif`. Directory Server 6.2 configuration is specified in the file `instance-path/config/dse.ldif`.

To migrate data from 5.1, perform the following steps:

1. Run the `migrateInstance5` migration script to produce a 5.2 configuration.
2. Use `dsmig` to migrate the 5.2 configuration.

You can directly use `dsmig` to migrate 5.1 configuration data.

For information on using `migrateInstance5`, see the *Directory Server 5.2 2005Q1 Installation and Migration Guide*. For information on using `dsmig` to migrate the configuration, see [“Using dsmig to Migrate Configuration Data” on page 30](#).

The following section describes the specific configuration attributes that must be migrated from the old instance to the new instance.

Migration of Specific Configuration Attributes

The values of the following attribute types must be migrated.

Global Configuration Attributes

The implementation of global scope ACIs requires all ACIs specific to the rootDSE to have a `targetscope` field, with a value of `base` (`targetscope="base"`). ACIs held in the rootDSE are specific to each Directory Server instance and are not replicated. Therefore there should be no incompatibility problems when running a Directory Server 6.2 server in a topology containing servers of previous versions. For more information about the changes made with regard to ACI scope, see [“Changes to ACIs” on page 68](#).

In addition to the ACI change, the following attributes under `cn=config` must be migrated:

```
nsslapd-accesscontrol
nsslapd-accesslog-level
nsslapd-accesslog-logbuffering
nsslapd-accesslog-logexpirationtime
nsslapd-accesslog-logexpirationtimeunit
nsslapd-accesslog-logging-enabled
nsslapd-accesslog-logmaxdiskpace
nsslapd-accesslog-logminfreediskspace
nsslapd-accesslog-logrotationtime
nsslapd-accesslog-logrotationtimeunit
nsslapd-accesslog-maxlogsize
```

nsslapd-accesslog-maxlogspedir
 nsslapd-attribute-name-exceptions
 nsslapd-auditlog-logexpirationtime
 nsslapd-auditlog-logexpirationtimeunit
 nsslapd-auditlog-logging-enabled
 nsslapd-auditlog-logmaxdiskspace
 nsslapd-auditlog-logminfreediskspace
 nsslapd-auditlog-logrotationtime
 nsslapd-auditlog-logrotationtimeunit
 nsslapd-auditlog-maxlogsize
 nsslapd-auditlog-maxlogspedir
 nsslapd-certmap-basedn
 nsslapd-ds4-compatible-schema
 nsslapd-enquote-sup-oc
 nsslapd-errorlog-level
 nsslapd-errorlog-logexpirationtime
 nsslapd-errorlog-logexpirationtimeunit
 nsslapd-errorlog-logging-enabled
 nsslapd-errorlog-logmaxdiskspace
 nsslapd-errorlog-logminfreediskspace
 nsslapd-errorlog-logrotationtime
 nsslapd-errorlog-logrotationtimeunit
 nsslapd-errorlog-maxlogsize
 nsslapd-errorlog-maxlogspedir
 nsslapd-groupevalnestlevel
 nsslapd-idletimeout
 nsslapd-infolog-area
 nsslapd-infolog-level
 nsslapd-ioblocktimeout
 nsslapd-lastmod
 nsslapd-listenhost
 nsslapd-maxbersize
 nsslapd-maxconnections
 nsslapd-maxdescriptors
 nsslapd-maxpsearch
 nsslapd-maxthreadsperconn
 nsslapd-nagle
 nsslapd-readonly
 nsslapd-referral
 nsslapd-referralmode
 nsslapd-reservedescriptors
 nsslapd-return-exact-case
 nsslapd-rootpwstoragescheme
 nsslapd-schema-repl-useronly
 nsslapd-schemacheck
 nsslapd-search-tune
 nsslapd-securelistenhost
 nsslapd-security

```
nsslapd-sizelimit  
nsslapd-threadnumber  
nsslapd-timelimit  
ds-start-tls-enabled
```

Security Configuration Attributes

All attributes under "cn=encryption,cn=config" must be migrated.

If you are using certificate authentication or the secure port, the key file path and certificate database file path under "cn=encryption,cn=config" must be updated. The values of the following attributes must be migrated:

```
nsKeyfile  
nsCertfile
```

Feature Configuration Attributes

The values of the aci attributes under "cn=features,cn=config" must be migrated.

In addition, the values of all identity mapping attributes must be migrated.

Mapping Tree Configuration Attributes

All entries under "cn=mapping tree,cn=config" must be migrated.

The Netscape Root database has been deprecated in Directory Server 6.2. If your old instance made specific use of the Netscape Root database, the attributes under o=netscaperoot must be migrated. Otherwise, they can be ignored.

Replication Configuration Attributes

Before migrating replication configuration attributes, ensure that there are no pending changes to be replicated. You can use the `insync` command to do this.

In addition to the configuration attributes, all entries under `cn=replication,cn=config` must be migrated. You must manually update the host and port on all replication agreements to the new instance, as well as the path to the change log database (`nsslapd-changelogdir`).

The following sections list the replication configuration attributes that must be migrated:

Change Log Attributes

TABLE 3-1 Change Log Attribute Name Changes

Old Attribute Name	Directory Server 6.2 Attribute Name
nsslapd-changelogmaxage	dschangelogmaxage
nsslapd-changelogmaxentries	dschangelogmaxentries

In addition, these attributes must be moved from cn=changelog5, cn=config to cn=replica, cn=suffixname, cn=mapping tree, cn=config entries (for each suffix name).

Fractional Replication Configuration Attributes

If your topology uses fractional replication, the following attribute names must be changed.

TABLE 3-2 Fractional Replication Attribute Name Changes

Old Attribute Name	Directory Server 6.2 Attribute Name
dsFilterSPType == fractional_include	dsReplFractionalInclude
dsFilterSPType == fractional_exclude	dsReplFractionalExclude

Replica Configuration Attributes

The values of the following replica configuration attributes must be migrated:

ds5ReferralDelayAfterInit
nsDS5Flags
nsDS5ReplicaBindDN
nsDS5ReplicaId
nsDS5ReplicaLegacyConsumer
nsDS5ReplicaName
nsDS5ReplicaPurgeDelay
nsDS5ReplicaReferral
nsDS5ReplicaRoot
nsDS5ReplicaTombstonePurgeInterval
aci

The dschangelogmaxage and dschangelogmaxentries attributes are added to the replica entry.

Replication Agreement Configuration

The values of the following attributes must be migrated for each replication agreement:


```

description
ds5agreementEnable
ds5ReplicaTransportCompressionLevel
ds5ReplicaTransportGroupSize
ds5ReplicaTransportWindowSize
nsDS5ReplicaBindDN
nsDS5ReplicaBindMethod
nsDS5ReplicaCredentials
nsDS5ReplicaHost
nsDS5ReplicaPort
nsDS5ReplicaRoot
nsDS5ReplicaTimeout
nsDS5ReplicaTransportInfo
nsDS5ReplicaUpdateSchedule
aci

```

Issues can arise when you migrate the `nsDS5ReplicaCredentials` attribute. For more information, see [“Manual Reset of Replication Credentials” on page 51](#).

There is no `ds5PartialReplConfiguration` attribute in Directory Server 6.2. This attribute must be removed.

If you are using fractional replication, the `dsReplFractionalInclude` and `dsReplFractionalExclude` attributes are added for each replication agreement.

All attributes under `"cn=replication,cn=config"` are migrated.

Password Policy Configuration Attributes

Directory Server 6.2 implements a new password policy. For details on configuration of the new password policy, see Chapter 7, “Directory Server Password Policy,” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*. The attributes that define the password policy are stored in the entry `cn=Password Policy,cn=config`. Note that in Directory Server 5.1, password policy attributes were located directly under `cn=config`.

Directory Server 6.2 introduces the new `pwdPolicy` object class. The attributes of this object class replace the old password policy attributes. For a description of these new attributes see the `pwdPolicy(5dsoc)` man page.

By default, the new password policy is backward compatible with the old password policy. However, because backward compatibility is not guaranteed indefinitely, you should migrate to the new password policy as soon as is convenient for your deployment. For information about password policy compatibility, see [“Password Policy Compatibility” on page 73](#).

While Directory Server 6.2 automatically manages coexistence between new and old password policies and entry operational attributes during migration and subsequent operations, you need to migrate any applications that refer to the old password policy attributes. The following table provides a mapping of the legacy password policy configuration attributes to the new attributes.

TABLE 3-3 Mapping Between 5 and 6.2 Password Policy Attributes

Legacy Directory Server Attribute	Directory Server 6.2 Attribute
passwordMinAge	pwdMinAge
passwordMaxAge	pwdMaxAge
passwordExp	pwdMaxAge
passwordInHistory	pwdInHistory
passwordSyntax	pwdCheckQuality
passwordMinLength	pwdMinLength
passwordWarning	pwdExpireWarning
-	pwdGraceLoginLimit
passwordMustChange	pwdMustChange
passwordChange	pwdAllowUserChange
-	pwdSafeModify
passwordStorageScheme	passwordStorageScheme
passwordExpireWithoutWarning	-
passwordLockout	pwdLockout
passwordLockoutDuration	pwdLockoutDuration
passwordUnlock	pwdLockoutDuration
passwordMaxFailure	pwdMaxFailure
passwordResetFailureCount	pwdFailureCountInterval

SNMP Attributes

The entry cn=SNMP, cn=config does not exist in Directory Server 6.2. All attributes under this entry are therefore deprecated. For information about setting up SNMP in Directory Server 6.2, see “Setting Up SNMP for Directory Server” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

UniqueID Generator Configuration Attributes

The nsState attribute under cn=uniqueid generator, cn=config must be migrated.

Database Configuration Attributes

General database configuration attributes are stored under cn=config, cn=ldbm database, cn=plugins, cn=config. The following attributes must be migrated:

```

nsslapd-lookthroughlimit
nsslapd-allidsthreshold
nsslapd-cache-autosize
nsslapd-cache-autosize-split
nsslapd-cachesize
nsslapd-db-checkpoint-interval
nsslapd-db-circular-logging
nsslapd-db-durable-transactions
nsslapd-db-idl-divisor
nsslapd-db-locks
nsslapd-db-logbuf-size
nsslapd-db-logfile-size
nsslapd-db-page-size
nsslapd-db-transaction-batch-val
nsslapd-db-tx-max
nsslapd-dbncache
nsslapd-import-cachesize
nsslapd-exclude-from-export
nsslapd-disk-low-threshold
nsslapd-disk-full-threshold

```

Database-specific attributes are stored in entries of the form *cn=database instance name*, *cn=ldbm database*, *cn=plugins*, *cn=config*. The following attributes must be migrated:

```

nsslapd-suffix
nsslapd-cachesize
nsslapd-cachememsize
nsslapd-readonly
nsslapd-require-index

```

If your deployment uses the NetscapeRoot suffix, you must migrate the attributes under *cn=netscapeRoot*, *cn=ldbm database*, *cn=plugins*, *cn=config*. You must also replace the database location (*nsslapd-directory*) with the location of the new Directory Server 6 instance.

All default index configuration attributes must be migrated, except for system indexes. Default index configuration attributes are stored in the entry *cn=default indexes*, *cn=ldbm database*, *cn=plugins*, *cn=config*. Indexes for the NetscapeRoot database do not need to be migrated.

All index configuration attributes must be migrated, except for system indexes. Index configuration attributes are stored in entries of the sort *cn=index name*, *cn=index*, *cn=database instance name*, *cn=ldbm database*, *cn=plugins*, *cn=config*.

All attribute encryption configuration attributes must be migrated.

Chained Suffix Attributes

All chained suffix configuration attributes must be migrated. The following configuration attributes are common to all chained suffixes. These attributes are stored in the entry `cn=config,cn=chaining database,cn=plugins,cn=config`.

```
nsActivechainingComponents
nsTransmittedControls
```

The following configuration attributes apply to a default instance of a chained suffix. These attributes are stored in the entry `cn=default instance config, cn=chaining database,cn=plugins,cn=config`.

```
nsAbandonedSearchCheckInterval
nsBindConnectionsLimit
nsBindRetryLimit
nsBindTimeout
nsCheckLocalACI
nsConcurrentBindLimit
nsConcurrentOperationsLimit
nsConnectionLife
nsHopLimit
nsmaxresponsedelay
nsmaxtestresponsedelay
nsOperationConnectionsLimit
nsProxiedAuthorization
nsReferralOnScopedSearch
nsslapd-sizelimit
nsslapd-timelimit
```

Plug-In Configuration Attributes

If you have changed the configuration of any standard plug-in, you must update that configuration. You must also update the configuration of all custom plug-ins. At a minimum, you must recompile all custom plug-ins and add their configuration to the directory. For a detailed list of plug-in API changes, see Chapter 2, “Changes to the Plug-In API Since Directory Server 5.2,” in *Sun Java System Directory Server Enterprise Edition 6.2 Developer's Guide*.

The following sections describe the standard plug-ins whose configuration must be migrated if you have changed it.

7-Bit Check Plug-In

The configuration of this plug-in is stored under `cn=7-bit check,cn=plugins,cn=config`. The following attributes must be migrated:

```
nsslapd-pluginarg*
nsslapd-pluginenabled
```

Class of Service Plug-In

The configuration of this plug-in is stored under `cn=Class of Service,cn=plugins,cn=config`. The following attributes must be migrated:

```
nsslapd-pluginarg0
nsslapd-pluginenabled
```

DSML Frontend Plug-In

The configuration of this plug-in is stored under `cn=DSMLv2-SOAP-HTTP,cn=frontends,cn=plugins,cn=config`. The following attributes must be migrated:

```
ds-hdsml-port
ds-hdsml-iobuffersize
ds-hdsml-requestmaxsize
ds-hdsml-responsemsgsize
ds-hdsml-poolsize
ds-hdsml-poolmaxsize
ds-hdsml-clientauthmethod
ds-hdsml-rooturl
ds-hdsml-soapschemalocation
ds-hdsml-dsmlschemalocation
nsslapd-pluginenabled
```

Pass Through Authentication Plug-In

The configuration of this plug-in is stored under `cn=Pass Through Authentication,cn=plugins,cn=config`. The following attribute must be migrated:

```
nsslapd-pluginenabled
```

The `nsslapd-pluginarg*` attributes must be migrated only if you require the configuration for `o=netscapeRoot` to be migrated.

Password Synchronization Plug-In

The configuration of this plug-in is stored under `cn=pswsync,cn=plugins,cn=config`. The following attribute must be migrated:

```
nsslapd-pluginenabled
```

Referential Integrity Plug-In

The configuration of this plug-in is stored under `cn=Referential Integrity Postoperation,cn=plugins,cn=config`. The following attributes must be migrated:

```
nsslapd-pluginarg*  
nsslapd-pluginenabled
```

Retro Change Log Plug-In

The configuration of this plug-in is stored under `cn=Retro Changelog PlugIn,cn=plugins,cn=config`. The following attributes must be migrated:

```
nsslapd-changelogmaxage  
nsslapd-changelogmaxentries  
nsslapd-pluginarg*  
nsslapd-pluginenabled
```

UID Uniqueness Plug-In

The configuration of this plug-in is stored under `cn=UID Uniqueness,cn=plugins,cn=config`. The following attributes must be migrated:

```
nsslapd-pluginarg*  
nsslapd-pluginenabled
```

Migrating Security Settings Manually

When you migrate an instance manually, the order in which you perform the migration of the security and the migration of the configuration is different to when you migrate using `dsmig`. If you migrate the security settings by replacing the default Directory Server 6.2 certificate and key databases with the old databases, as described in this section, you *must* migrate the configuration first.

To migrate the security settings manually, perform the following steps:

1. If you have already started using the new instance, stop the instance.
2. Back up the certificate database and key database files on the new instance.
3. Copy the certificate database and key database files from the existing instance to the new instance.

```
$ cp serverRoot/alias/slapd-serverID-cert8.db instance-path/alias/slapd-cert8db  
$ cp serverRoot/alias/slapd-serverID-key3.db instance-path/alias/slapd-key3.db
```

For 5.1 servers and earlier releases of 5.2 servers, the certificate database to be copied is `serverRoot/alias/slapd-serverID-cert7.db`.

4. Copy the password file from the existing instance to the new instance.

```
$ cp serverRoot/alias/slapd-serverID-pin.txt instance-path/alias/slapd-pin.txt
```

5. Update the certificate database password.

```
$ dsadm set-flags instance-path cert-pwd-prompt=on
```

6. Copy the certificate mapping file from the existing instance to the new instance.

```
$ cp serverRoot/shared/config/certmap.conf instance-path/alias/certmap.conf
```

7. If the existing instance uses an external security token, copy the security module database and the external token library to the new instance.

```
$ cp serverRoot/alias/secmod.db instance-path/alias/secmod.db
```

8. Start the new instance.

The security configuration attributes are migrated when you migrate the rest of the configuration attributes. In this sense, migration of the security settings is not complete until you have migrated the configuration. Migration of the configuration is described in the following section.

Migrating User Data Manually

If your topology does not support automatic data migration, you must migrate the data manually. This involves exporting the data from the existing instance and re-importing it to the new instance.

To migrate data manually from an existing version 5 instance, perform the following steps:

1. If you already have data in the new instance, back up any conflicting suffixes in the new instance.
2. If you are migrating a master server instance in a replicated topology, make sure that the master is synchronized with all servers that are direct consumers of that master.
It is not possible to migrate the change log manually. A new change log is created in the 6.2 instance.
3. Export the required suffixes to LDIF by using the `db2ldif` command. This command exports all the suffix contents to an LDIF file, when the server is either running or stopped.
The following example exports two suffixes to a single LDIF file.

```
$ serverRoot/slapd-serverID/db2ldif -a example.ldif \  
-r -s "ou=people,dc=example,dc=com" -s "ou=departments,dc=example,dc=com"
```

In this example, `-a` specifies the resulting LDIF file, `-r` indicates that replication information should be exported, and `-s` specifies the suffixes to be included in the export.

4. On the new instance, import the LDIF files by using the `dsadm import` command. For example, the following commands import the LDIF file created previously into the two suffixes that were exported.

```
$ dsadm import instance-path example.ldif ou=people,dc=example,dc=com  
$ dsadm import instance-path example.ldif ou=departments,dc=example,dc=com
```

5. If the retro change log was configured on the 5.2 instance, export the retro change log to LDIF by using the `db2ldif` command.

```
$ serverRoot/slapd-serverID/db2ldif -a changelog.ldif \  
-s "cn=changelog"
```

In this example, `-a` specifies the resulting LDIF file, and `-s` specifies the changelog suffix.

6. On the new instance, import the retro change log using the `dsadm import` command. For example, the following command imports the change log LDIF file created previously.

```
$ dsadm import instance-path changelog.ldif cn=changelog
```

7. Start the new instance.

Note – During data migration, Directory Server checks whether nested group definitions exceed 30 levels. Deep nesting can signify a circular group definition, where a nested group contains a group that is also its parent. When a group with more than 30 nesting levels is encountered, Directory Server stops calculating the `isMemberOf` attributes for additional levels.

Each time this happens, Directory Server logs an error. You safely ignore these errors, although you should examine the definition of the group mentioned in the error message for potential circular definitions.

Migrating User Plug-Ins Manually

User plug-ins cannot be migrated. If you have custom user plug-ins, recompile them and add them to the Directory Server 6.2 instance manually. For a detailed list of plug-in API changes, see Chapter 2, “Changes to the Plug-In API Since Directory Server 5.2,” in *Sun Java System Directory Server Enterprise Edition 6.2 Developer’s Guide*.

Tasks to be Performed After Manual Migration

If you have migrated your server manually, the following post-migration tasks are required before you can run the new server.

- If you have customized user plug-ins, these need to be recompiled and added to the new server manually.
- If the migrated server was part of a replicated topology, see [Chapter 4, “Migrating a Replicated Topology.”](#)
- If you have customized backup, recovery, and installation scripts, you need to rewrite these scripts to comply with the new version.

Migrating a Replicated Topology

Directory Server Enterprise Edition 6.2 does not provide a way to migrate an entire replicated topology automatically. Migrating a replicated topology involves migrating each server individually. Usually, however, you should be able to migrate your entire topology without any interruption in service.

This chapter describes the issues involved in migrating replicated servers, and covers the following topics:

- [“Overview of Migrating Replicated Servers” on page 49](#)
- [“Issues Related to Migrating Replicated Servers” on page 50](#)
- [“New Replication Recommendations” on page 51](#)
- [“Migration Scenarios” on page 52](#)

Overview of Migrating Replicated Servers

Directory Server 6.2 supports an unlimited number of masters in a multi-master topology. This and other changes might mean that you redesign your topology rather than migrate to an identical topology with new servers. See Part III, “Logical Design,” in *Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide* before continuing.

When migrating replicated version 5 servers, you typically start with the consumers, continue with the hubs, and finish with the masters. This bottom-up approach involves interrupting only one server at a time, rather than interrupting an entire branch of the replication topology. The approach also helps you avoid potential custom schema synchronization issues between masters and consumers.

Issues Related to Migrating Replicated Servers

Depending on your replication topology, and on your migration strategy, certain issues might arise when you migrate replicated servers. These issues are described in the following sections.

Issues With the New Password Policy

If you are migrating a multi-master replicated topology, a situation will arise where a 6.2 master is replicating to a version 5 server. In this situation, an object class violation will occur if changes are made to the new password policy attributes on the 6.2 server, and replicated to the version 5 server. The password policy attributes are managed internally by the server but they might be updated in the event of a bind, a user password modify, or the addition of an entry with the `userpassword` attribute.

To avoid the object class violation, the 6.2 password policy schema file (`00ds6pwp.ldif`) *must* be copied to every version 5 server that will be supplied by a 6.2 master. When the password policy schema file has been copied, restart the version 5 server.

Migration of Replication Agreements

If possible, you should migrate replicated servers to the same host name and port number. If you *must* change the host name or port number of a replicated server, all replication agreements that point to that server must be updated manually to point to the new server. For example, if you migrate a consumer server from `red.example.com:1389` to `blue.example.com:1389`, the replication agreements on all masters that point to `red.example.com:1389` must be updated manually to point to `blue.example.com:1389`.

Replication agreements *from* the migrated master to consumers in the topology are managed by the `dsrmig` migration tool. If your topology does not support automated migration, these replication agreements must also be updated manually.

Migration of Referrals

Referrals are also affected if you migrate a *master* replica to a new host or port. The details of each master in a topology are present in the Replica Update Vector (RUV) of all other servers in the topology. The RUV of each server is used to determine the referrals. When you change the host name or port number of a master server during migration, all referrals to that master from other servers in the topology become invalid. The easiest way to correct this is to use the following steps, in order, when performing the migration.

1. Before migrating a master server, verify that there are no pending changes to be replicated. You can use the `insync` tool to do this.

2. Demote the master server to a hub, as described in “Promoting or Demoting Replicas” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.
3. Migrate the hub server, either using `dsmig` or the manual migration progress.
4. Promote the hub server to a master, as described in “Promoting or Demoting Replicas” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*. When you promote the hub, you must assign a `replicaID` to the new migrated master. This new `replicaID` must be different to the `replicaID` of the old server that is being migrated, and must be unique within the replicated topology.

Manual Reset of Replication Credentials

`dsmig` does not migrate the password of the default replication manager entry (`cn=replication manager,cn=replication,cn=config`). Instead, the replication manager password is deleted. Therefore, whether you are using manual or automatic migration, you must reset the replication manager password manually.

To reset the replication manager password, use the following command:

```
$ dsconf set-server-prop -h host -p port def-repl-manager-pwd-file:filename
```

In addition, `dsmig` does not migrate non-default replication manager entries. If a version 5 replica uses an entry other than the default replication manager, and if this entry is under `cn=config`, you must add the default replication manager manually. Please refer to the documentation to add a non-default replication manager entry manually. For information about adding a non-default replication manager, see “Using a Non-Default Replication Manager” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

Problems Related to Tombstone Purging

In some cases, after migrating a replicated topology you might experience problems related to tombstone purging. In some cases, tombstone entries are not purged when they should be. This problem can be resolved by re-indexing the `objectclass` attribute of the corresponding suffix.

New Replication Recommendations

Directory Server 6.2 does not limit the number of masters in a multi-master topology. A fully-meshed, multi-master topology with no hubs or consumers is recommended in most cases.

Advantages of an all-master topology include the following:

- **Availability.** Write traffic is never disrupted if one of the servers goes down.
- **Simplicity.** In an all-master topology, there is no need to set up referrals to route reads and writes to different servers.

There may be reasons that an all-master topology is not viable in a specific deployment. For example, fractional replication cannot be used in an all-master topology because fractional replication is only supported from masters to consumers.

Migration Scenarios

This section provides sample migration scenarios for a variety of replicated topologies.

Migrating a Replicated Topology to an Identical Topology

Before you start migrating replicated servers, determine whether your deployment might not be better served by changing the architecture of the topology. This section describes how to migrate if you want to keep your existing topology. Migrating a replicated topology to an identical topology, involves migrating the consumers, then the hubs, then the masters. The following sections demonstrate a sample migration of a simple multi-master topology.

Migrating the Consumers

For each consumer in the replicated topology:

1. Reroute clients to another consumer in the topology.
2. Disable any replication agreements to the consumer you want to migrate.
3. Stop the consumer.
4. Migrate the consumer according to the instructions under [Chapter 1, “Overview of the Migration Process for Directory Server.”](#)
5. Start the consumer.
6. Enable the replication agreements from the hubs to that consumer.
7. If you have migrated the data, check that replication is in sync.
8. If you have not migrated the data, reinitialize the consumer.
9. Reroute clients back to the consumer.

The following sequence of diagrams illustrate the migration of a consumer, as described above. The first diagram shows the version 5 topology before the migration.

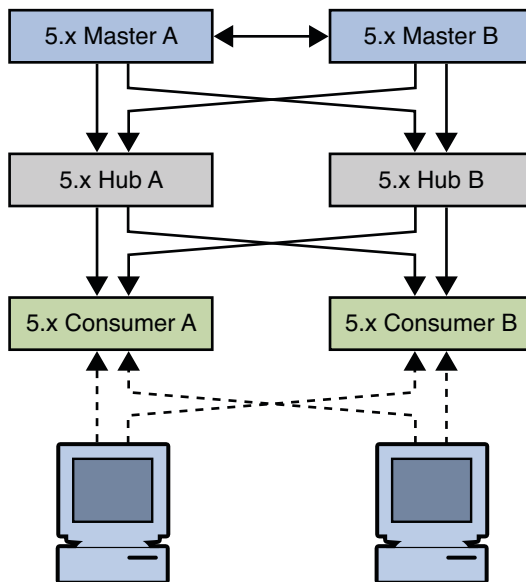


FIGURE 4-1 Existing version 5 Topology

The first step involves rerouting clients and disabling replication agreements, effectively isolating the consumer from the topology.

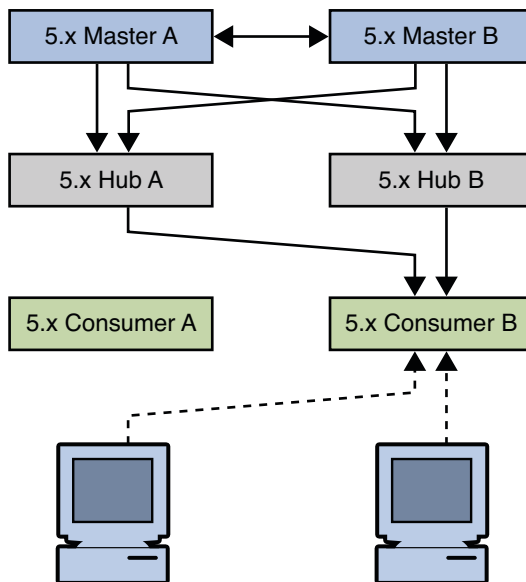


FIGURE 4-2 Isolating the Consumer From the Topology

The next step involves migrating the version 5 consumer.

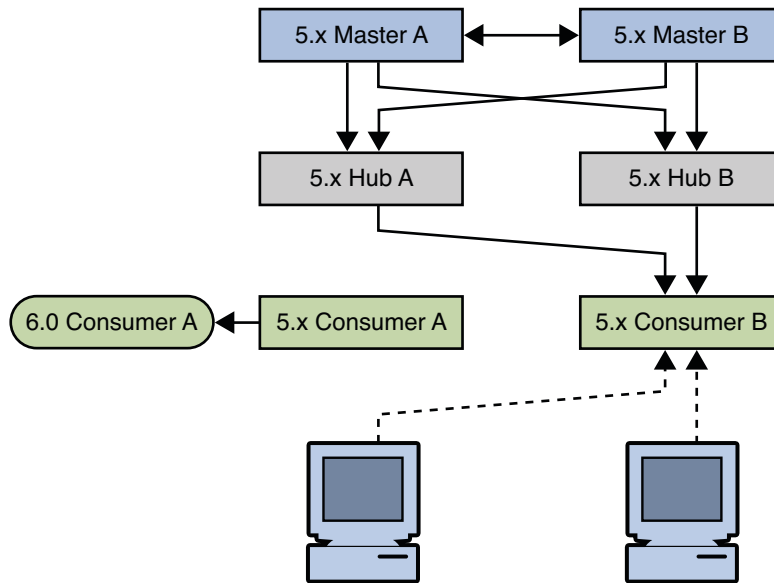


FIGURE 4-3 Migrating the version 5 Consumer

The next step involves enabling the replication agreements to the new consumer, initializing the consumer if necessary, and rerouting client applications to the new consumer.

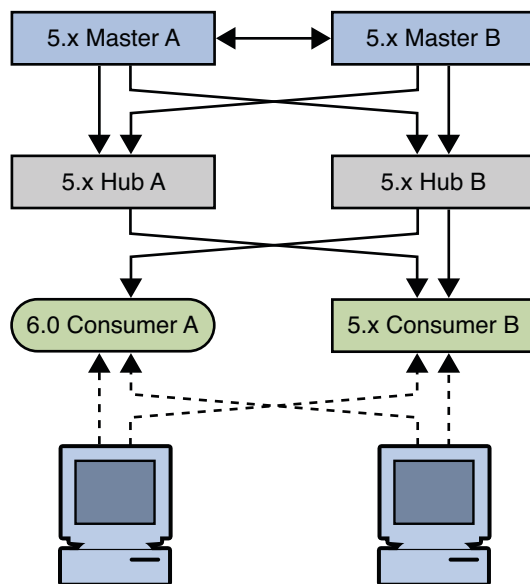


FIGURE 4-4 Placing the 6.0 Consumer Into the Topology

Migrating the Hubs

For each hub in the replicated topology:

1. Disable replication agreements from the masters to the hub you want to migrate.
2. Disable replication agreements from the hub you want to migrate to the consumers.
3. Stop the hub.
4. Migrate the hub according to the instructions under [Chapter 1, “Overview of the Migration Process for Directory Server.”](#)
5. Start the hub.
6. Enable the replication agreements from the masters to that hub.
7. Enable the replication agreements from that hub to the consumers.
8. If you have migrated the data, check that replication is in sync.
9. If you have not migrated the data, reinitialize the hub.

The following sequence of diagrams illustrate the migration of a hub, as described above. The first diagram shows the topology before migrating the hubs.

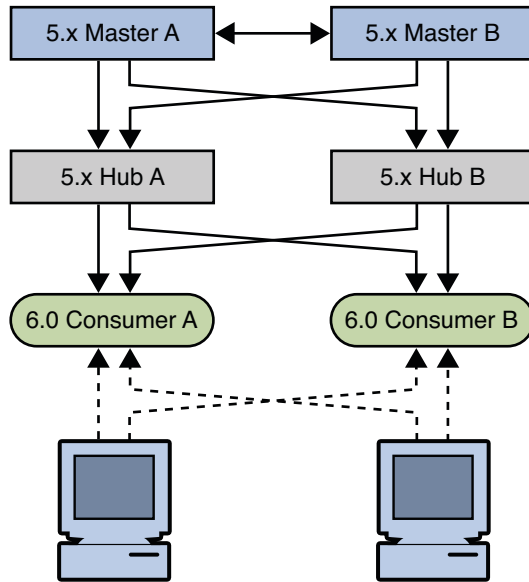


FIGURE 4-5 Existing version 5 Topology With Migrated Consumers

The first migration step involves disabling replication agreements, effectively isolating the hub from the topology.

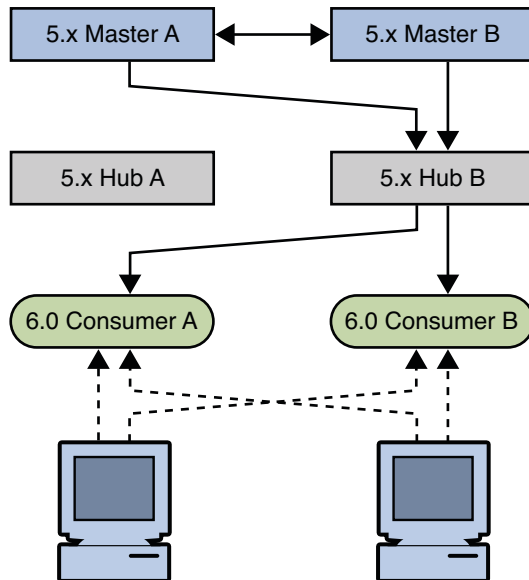


FIGURE 4-6 Isolating the Hub From the Topology

The next step involves migrating the version 5 hub.

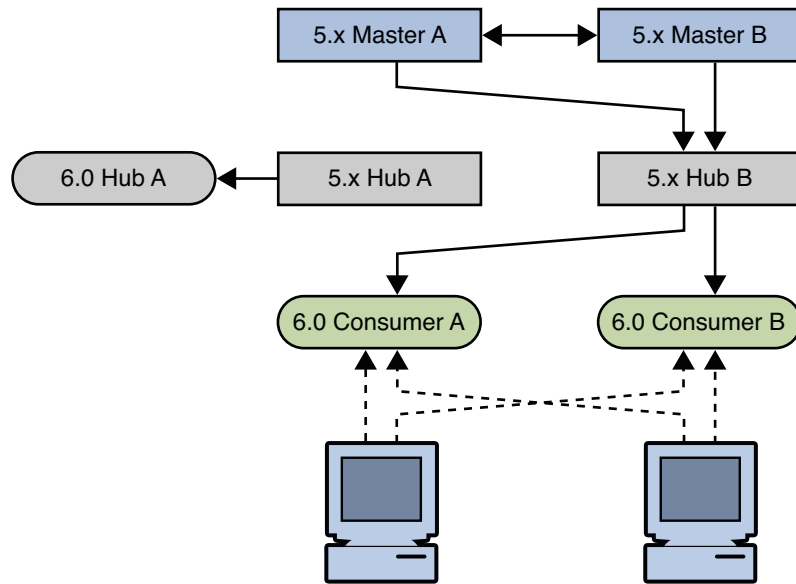


FIGURE 4-7 Migrating the version 5 Hub

The next step involves enabling the replication agreements to the new hub and initializing the hub if necessary.

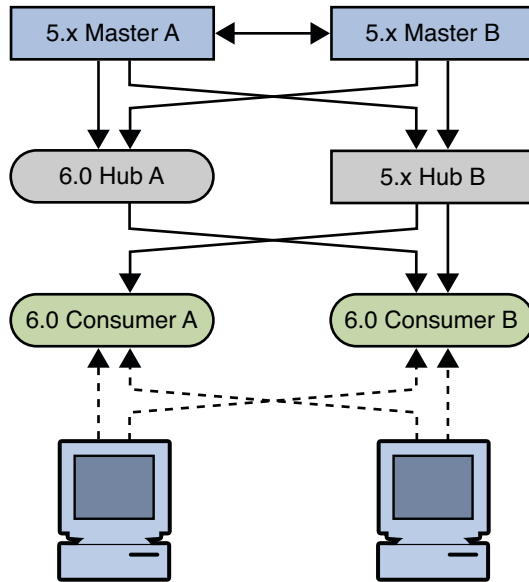


FIGURE 4-8 Placing the 6.0 Hub Into the Topology

Check that the replication on the consumers is in sync with the rest of the topology before migrating another hub. A server that has just been migrated does not have a change log, and can therefore not update consumer servers that are out of sync. Allow the topology to stabilize and all servers to synchronize before migrating the next supplier server.

Migrating the Masters

For each master in the replicated topology:

1. If you have client applications that write to the master you want to migrate, reroute these applications to write to another master in the topology.
2. Ensure that the master is no longer receiving write requests. You can do this by enabling read-only mode on the master.
3. Check that replication is synchronized between the master and all its consumers.

Migration of the change log is not supported if you are migrating manually, so the preceding two steps are mandatory in this case. Although automatic migration *does* migrate the change log, you should still perform the above steps to avoid the risk of losing changes.

4. Disable any replication agreements to and from the master you want to migrate.
5. Stop the master.
6. Migrate the master according to the instructions under [Chapter 1, “Overview of the Migration Process for Directory Server.”](#)
7. Start the master.

8. Enable the replication agreements from the master to the hubs and other masters in the topology.
9. If you have migrated the data, check that replication is in sync.
10. If you have not migrated the data, reinitialize the master from another master in the topology.
11. If you rerouted client applications (Step 2), you can now route the applications to write to the migrated master.

The following sequence of diagrams illustrate the migration of a master, as described above. The first diagram shows the version 5 topology before the migration of the masters.

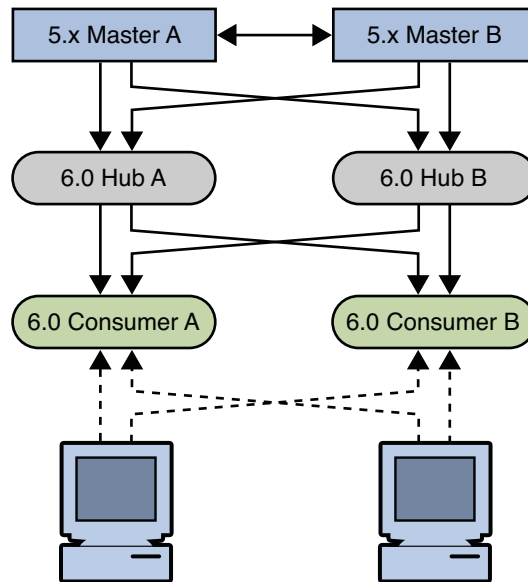


FIGURE 4-9 Existing version 5 Topology With Consumers and Hubs Migrated

The first step in migrating a master involves disabling replication agreements, effectively isolating the master from the topology.

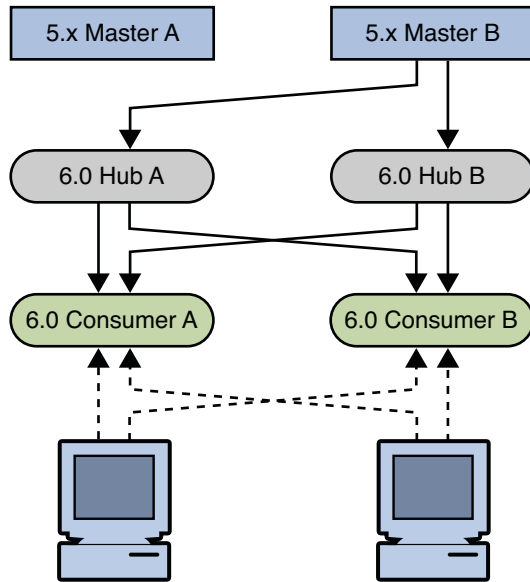


FIGURE 4-10 Isolating the Master From the Topology

The next step involves migrating the version 5 master.

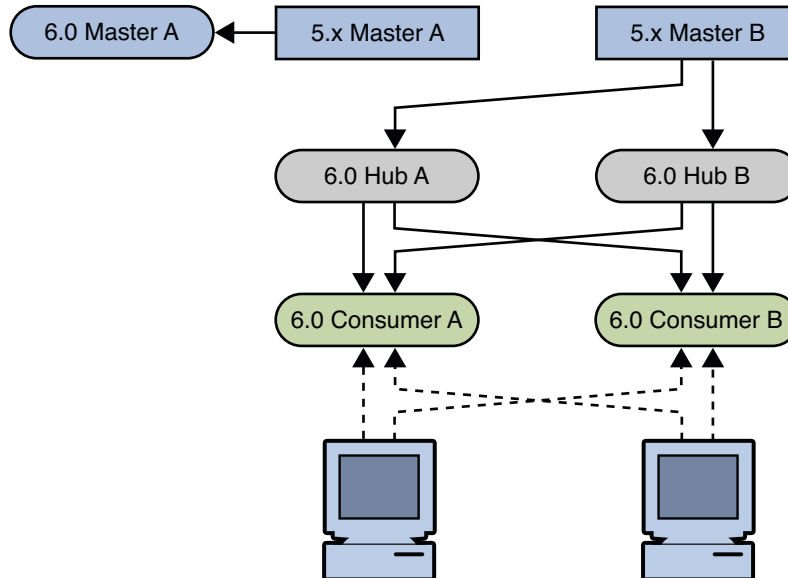


FIGURE 4-11 Migrating the version 5 Master

The next step involves enabling the replication agreements to and from the new master and initializing the master if necessary.

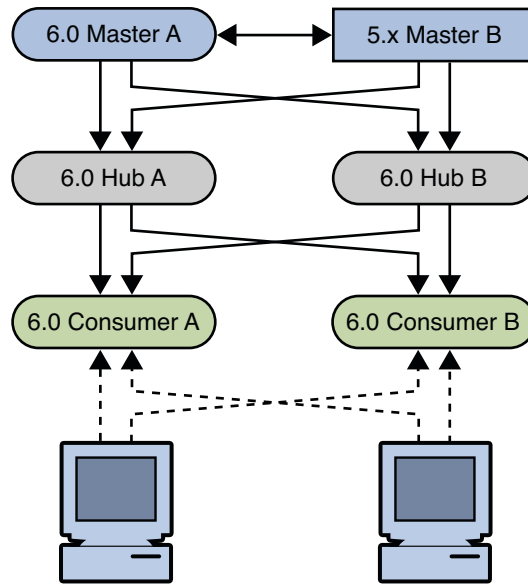


FIGURE 4-12 Placing the 6.0 Master Into the Topology

Check that the replication on all hubs and consumers is in sync with the rest of the topology before migrating another master. A server that has just been migrated does not have a change log, and can therefore not update servers that are out of sync. Allow the topology to stabilize and all servers to synchronize before migrating the next supplier server.

Migrating a Replicated Topology to a New Topology

Before you start migrating replicated servers, determine whether your deployment might not be better served by changing the architecture of the topology. This section describes how to migrate a basic version 5 topology to a new all-master topology. Migrating to an all-master topology involves migrating the consumers, hubs, and masters, then promoting the hubs to masters and the consumers to hubs, then to masters. The following sections demonstrate a sample migration of a simple multi-master topology to a new all-master topology.

The following figure shows the existing version 5 topology.

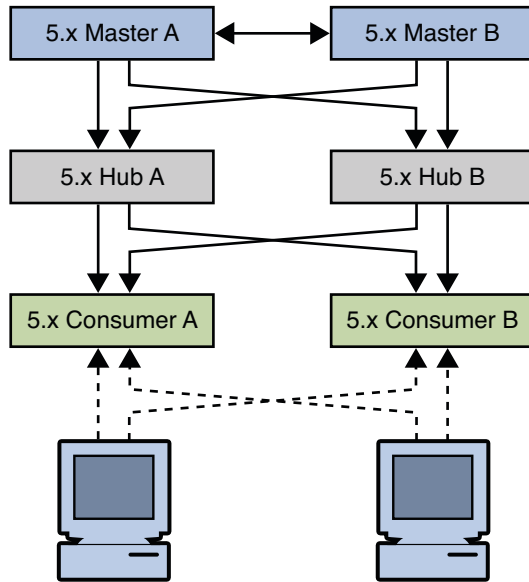


FIGURE 4-13 Existing version 5 Topology

Migrating All the Servers

The first step is to migrate all the servers individually, as described in [“Migrating a Replicated Topology to an Identical Topology” on page 52](#). The resulting topology is illustrated in the following figure.

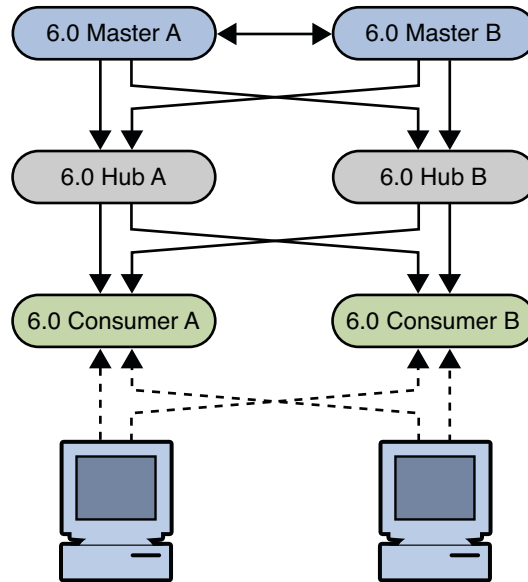


FIGURE 4-14 Existing Topology With Migrated Servers

Promoting the Hubs

The next step involves promoting the hubs to masters, and creating a fully-meshed topology between the masters. To promote the hubs, follow the instructions in “Promoting or Demoting Replicas” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

The following diagram illustrates the topology when the hubs have been promoted.

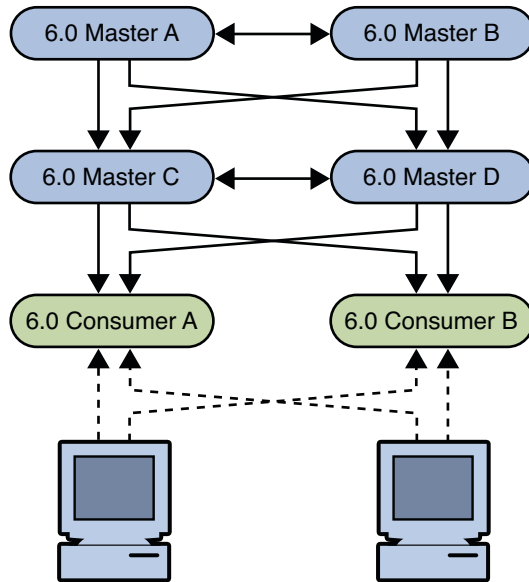


FIGURE 4-15 Migrated Topology With Promoted Hub Replicas

Promoting the Consumers

The next step involves promoting the consumers to hubs, and then to masters, and creating a fully-meshed topology between the masters. To promote the consumers, follow the instructions in “Promoting or Demoting Replicas” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

The following diagram illustrates the topology when the consumers have been promoted.

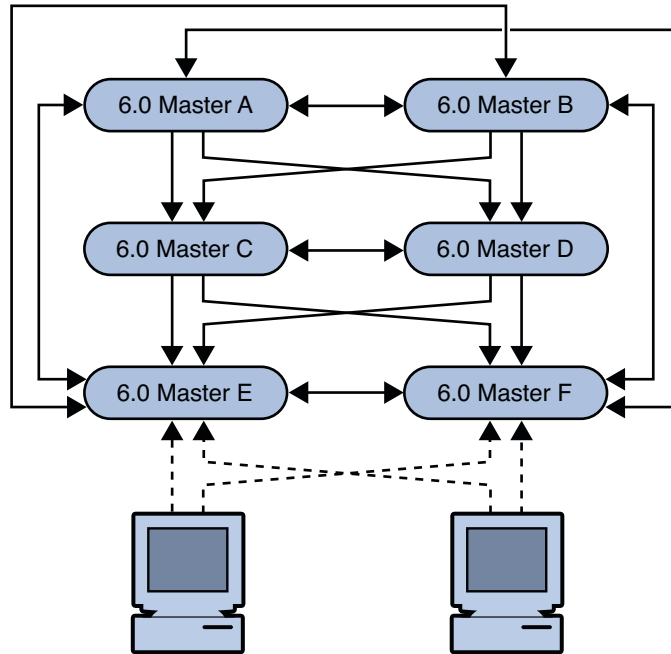


FIGURE 4-16 New Fully-Meshed All-Master Topology

Migrating Over Multiple Data Centers

Migrating servers over multiple data centers involves migrating each server in each data center individually. Before you start migrating replicated servers, determine whether your deployment might not be better served by changing the architecture of the topology. If you want to keep your existing topology, follow the examples in [“Migrating a Replicated Topology to an Identical Topology” on page 52](#) for each data center. To migrate to a new topology, follow the examples in [“Migrating a Replicated Topology to a New Topology” on page 61](#) for each data center.

Architectural Changes in Directory Server

This chapter describes the architectural changes in Directory Server that affect migration from a previous version. For information on *all* changes and bug fixes in Directory Server, see “What’s New at a Glance” in *Sun Java System Directory Server Enterprise Edition 6.2 Evaluation Guide*.

This chapter covers the following topics:

- “Changes in the Administration Framework” on page 67
- “Changes to ACIs” on page 68
- “Command Line Changes” on page 69
- “Changes to the Console” on page 72
- “New Password Policy” on page 72
- “Changes to Plug-Ins” on page 78
- “Changes to the Installed Product Layout” on page 79

Changes in the Administration Framework

Directory Server 6.2 does not include an administration server, as in previous versions. Servers are now registered in the Directory Service Control Center (DSCC) and can be administered remotely by using the web-based GUI or the command-line tools.

To migrate to the new administration framework, you need to do the following:

- Upgrade each server individually
- Register each server in the DSCC

Removal of the *ServerRoot* Directory

In the new administration model, a Directory Server instance is no longer tied to a *ServerRoot*. Each Directory Server instance is a standalone directory that can be manipulated in the same manner as an ordinary standalone directory.

Removal of the o=netscapeRoot Suffix

In previous versions of Directory Server, centralized administration information was kept in o=netscapeRoot. In the new administration model, the concept of a *configuration directory server* no longer exists. The o=netscapeRoot suffix is no longer required, and the netscapeRoot database files are therefore *not* migrated. The configuration data for this suffix can be migrated, if it is specifically required.

Changes to ACIs

The following changes have been made to ACIs in Directory Server 6.2.

Changes in the ACI Scope

In Directory Server 5.2 ACIs on the root DSE had base scope. In Directory Server 6.2, ACIs on the root DSE have global scope by default, equivalent to targetscope="subtree".

To reproduce the same behavior as Directory Server 5.2, add targetscope="base" to ACIs on the root DSE. If you use dsmig to migrate the configuration, this is done automatically.

Changes in Suffix-Level ACIs

In Directory Server 5.2, the following ACI was provided, at the suffix level:

```
aci: (targetattr != "nsroledn || aci || nsLookThroughLimit ||
    nsSizeLimit || nsTimeLimit || nsIdleTimeout || passwordPolicySubentry ||
    passwordExpirationTime || passwordExpWarned || passwordRetryCount ||
    retryCountResetTime || accountUnlockTime || passwordHistory ||
    passwordAllowChangeTime")(version 3.0; acl "Allow self entry modification
except for nsroledn, aci, resource limit attributes, passwordPolicySubentry
and password policy state attributes"; allow (write)userdn = "ldap:///self");
```

This ACI allowed self-modification of user passwords, among other things. This ACI is no longer provided in Directory Server 6.2. Instead, the following global ACIs are provided by default:

```
aci: (targetattr != "aci") (targetscope = "base") (version 3.0;
aci "Enable read access to rootdse for anonymous users";
allow(read,search,compare) user dn="ldap:///anyone"; )
```

```
aci: (targetattr = "*") (version 3.0; acl "Enable full access
for Administrators group"; allow (all)(groupdn =
"ldap:///cn=Administrators,cn=config"); )
```

```
aci: (targetattr = "userPassword") ( version 3.0; acl "allow
userpassword self modification"; allow (write) userdn = "ldap:///self");)
```

In Directory Server 6.2, the default userPassword ACI at root DSE level provides equivalent access control to the default 5.2 ACI at suffix level. However, if you want to reproduce exactly the same access control as in 5.2, add the following ACI to your suffix. This ACI is the 5.2 ACI, with the new password policy operational attributes for Directory Server 6.2.

```
aci: (targetattr != "nsroledn || aci || nsLookThroughLimit ||
nsSizeLimit || nsTimeLimit || nsIdleTimeout || passwordPolicySubentry ||
passwordExpirationTime || passwordExpWarned || passwordRetryCount ||
retryCountResetTime || accountUnlockTime || passwordHistory ||
passwordAllowChangeTime || pwdAccountLockedTime || pwdChangedTime ||
pwdFailureTime || pwdGraceUseTime || pwdHistory ||
pwdLastAuthTime || pwdPolicySubentry || pwdReset")(version 3.0; acl "Allow self entry modification
except for nsroledn, aci, resource limit attributes, passwordPolicySubentry
and password policy state attributes"; allow (write)userdn ="ldap:///self";)
```

Tip – Do not allow users write access to everything and then deny write access to specific attributes. Instead, explicitly list the attributes to which you allow write access.

Command Line Changes

In Directory Server 6.2 the functionality of most command-line tools is replaced by only two commands: `dsadm` and `dsconf`.

The following table shows commands used in Directory Server 5, and the corresponding commands for Directory Server 6.2. The default path of these commands when installed from native packages is `/opt/SUNWdsee/ds6/bin`. When installed from the zip installation, the default path is *install-path/ds6/bin*.

TABLE 5-1 Directory Server 5 and 6 commands

Version 5 Command	Version 6.2 Command	Description
bak2db	dsadm restore	Restore a database from backup (locally, offline)
bak2db-task	dsconf restore	Restore a database from backup (remotely, online)
db2bak	dsadm backup	Create a database backup archive (locally, offline)

TABLE 5-1 Directory Server 5 and 6 commands *(Continued)*

Version 5 Command	Version 6.2 Command	Description
db2bak-task	dsconf backup	Create a database backup archive (remotely, online)
db2index	dsadm reindex	Create and generate indexes (locally, offline)
db2index-task	dsconf reindex	Create and generate indexes (remotely, online)
db2ldif	dsadm export	Export database contents to LDIF (locally, offline)
db2ldif-task	dsconf export	Export database contents to LDIF (remotely, online)
entrycmp	No change	Compare the same entry in multiple replicas
fildif	No change	Create a filtered version of an LDIF file
idsktune	No change	Check patches and verifies system tuning
insync	No change	Indicate synchronization between multiple replicas
ldif2db	dsadm import	Import database contents from LDIF (locally, offline)
ldif2db-task	dsconf import	Import database contents from LDIF (remotely, online)
ldif2ldap	ldapmodify -B	Import data from LDIF over LDAP (remotely, online)
MigrateInstance5	dsmig / manual migration procedure	Migrate data from a previous version
mmldif	No change	Combine multiple LDIF files
monitor	ldapsearch on cn=monitor	Retrieve performance monitoring information
pwdhash	No change	Print the encrypted form of a password
repldisc	No change	Discover a replication topology
restart-slapd	dsadm restart	Restart a Directory Server instance
schema_push	No change	Update schema modification time stamps
start-slapd	dsadm start	Start a Directory Server instance

TABLE 5-1 Directory Server 5 and 6 commands (Continued)

Version 5 Command	Version 6.2 Command	Description
stop-slapd	dsadm stop	Stop a Directory Server instance
suffix2instance	dsconf get-suffix-prop	See the backend name for a suffix
vlvindex	dsadm reindex	Create virtual list view indexes

TABLE 5-2 Directory Server 5 and 6 Commands (Subcommands of the `directoryserver` Command)

Version 5 Command	Version 6.2 Command	Description
directoryserver accountstatus	ns-accountstatus	Establish account status
directoryserver activate	ns-activate	Activate an entry or group of entries
directoryserver configure	Installation procedure	Install Directory Server
directoryserver inactivate	ns-inactivate	Inactivate an entry or group of entries
directoryserver unconfigure	Uninstallation procedure	Uninstall Directory Server

Deprecated Commands

Some version 5 commands have been deprecated in Directory Server 6.2. The following table provides a list of these commands.

TABLE 5-3 Version 5 Commands That Have Been Deprecated

Command	Description
getpwenc	Print encrypted password
ns-ldapagt	Starts a Directory Server SNMP subagent. For information about how to do this in Directory Server 6.2, see “To Set Up SNMP” in <i>Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide</i>
restore-config	Restore Administration Server configuration
saveconfig	Save Administration Server configuration

Changes to the Console

The downloaded, Java Swing-based console has been replaced by Directory Service Control Center (DSCC). DSCC is a graphical interface that enables you to manage an entire directory service by using a web browser. The DSCC requires no migration. Migrated Directory Server instances can be registered in the DSCC. For more information about the DSCC see Chapter 1, “Directory Server Overview,” in *Sun Java System Directory Server Enterprise Edition 6.2 Reference*.

New Password Policy

Directory Server 6.2 implements a new password policy that uses the standard object class and attributes described in the “[Password Policy for LDAP Directories](#)” Internet-Draft.

The new password policy provides the following new features:

- A grace login limit, specified by the `pwdGraceAuthNLimit` attribute. This attribute specifies the number of times an expired password can be used to authenticate. If it is not present or if it is set to 0, authentication will fail.
- Safe password modification, specified by the `pwdSafeModify` attribute. This attribute specifies whether the existing password must be sent when changing a password. If the attribute is not present, the existing password does not need to be sent.

In addition, the new password policy provides the following new controls:

- `LDAP_CONTROL_PWP_[REQUEST|RESPONSE]`
- `LDAP_CONTROL_ACCOUNT_USABLE_[REQUEST|RESPONSE]`

These controls enable LDAP clients to obtain account status information.

The `LDAP_CONTROL_PWP` control provides account status information on LDAP bind, search, modify, add, delete, modDN, and compare operations.

The following information is available, using the OID `1.3.6.1.4.1.1.42.2.27.8.5.1` in the search:

- Period of time before the password expires
- Number of grace login attempts remaining
- The password has expired
- The account is locked
- The password must be changed after being reset
- Password modifications are allowed
- The user must supply his/her old password
- The password quality (syntax) is insufficient
- The password is too short

- The password is too young
- The password already exists in history

The LDAP_CONTROL_PWP control indicates warning and error conditions. The control value is a BER octet string, with the format {t i i}, which has the following meaning:

- t is a tag defining which warning is set, if any. The value of t can be one of the following:

```
LDAP_PWP_WARNING_RESP_NONE (0x00L)
LDAP_PWP_WARNING_RESP_EXP (0x01L)
LDAP_PWP_WARNING_RESP_GRACE (0x02L)
```

- The first i indicates warning information. The warning depends on the value set for t as follows:
 - If t is set to LDAP_PWP_WARNING_RESP_NONE, the warning is - 1.
 - If t is set to LDAP_PWP_WARNING_RESP_EX, the warning is the number of seconds before expiration.
 - If t is set to LDAP_PWP_WARNING_RESP_GRACE, the warning is the number of remaining grace logins.
- The second i indicates error information. If t is set to LDAP_PWP_WARNING_RESP_NONE, the error contains one of the following values:

```
pwp_resp_no_error (-1)
pwp_resp_expired_error (0)
pwp_resp_locked_error (1)
pwp_resp_need_change_error (2)
pwp_resp_mod_not_allowed_error (3)
pwp_resp_give_old_error (4)
pwp_resp_bad_qa_error (5)
pwp_resp_too_short_error (6)
pwp_resp_too_young_error (7)
pwp_resp_in_hist_error (8)
```

The LDAP_CONTROL_ACCOUNT_USABLE control provides account status information on LDAP search operations only.

Password Policy Compatibility

For migration purposes, the new password policy maintains compatibility with previous Directory Server versions by implementing a compatibility mode. The compatibility mode determines whether password policy attributes are handled as *old* attributes or *new* attributes, where *old* refers to Directory Server 5 password policy attributes.

This section contains information to help you set the compatibility mode and to decide which mode is appropriate for your deployment.

Setting the Compatibility Mode

The compatibility mode can be read using `dsconf` command as follows:

```
$ dsconf get-server-prop pwd-compat-mode
```

The `pwd-compat-mode` property can have one of the following values:

- | | |
|---------------------|---|
| DS5-compatible-mode | The purpose of DS5-compatible-mode is to allow an existing replicated topology to be migrated from Directory Server 5 instances to Directory Server 6.2 instances. A Directory Server 6.2 instance in DS5-compatible-mode accepts updates containing either old or new password policy attributes, and produces updates containing both sets of attributes. Updates can arrive from an LDAP client or from replication, and changes are produced locally as a result of password policy evaluation (for example, as a result of a failed authentication or a password change). Note that any version 5 instance consuming replicated updates produced by a version 6.2 instance (either directly or through another instance) must have its schema updated with <code>00ds6pwp.ldif</code> as described in “Issues With the New Password Policy” on page 50 . While the Directory Server 5 instance will ignore any new attributes during password policy processing, when an entry containing the new attributes is modified at that instance, without the schema update, the schema check will fail when the modified entry is written. |
| DS6-migration-mode | DS6-migration-mode is an intermediate step between DS5-compatible-mode and DS6-mode. All policy decisions are based on the new password policy attributes and the old (Directory Server 5) password policy attributes are removed from the server's data. Since the number of policy configuration entries is small, the old password policy configuration attributes are cleaned from all policy entries as soon as the instance is advanced to DS6-migration-mode. However, the cleanup of a user entry is deferred until the entry is modified as part of normal password policy processing during a password modify operation. This approach allows the cleanup to proceed gradually, so as to not degrade the server's performance. The modifications necessary to remove any old policy attributes are not replicated so that they do not interfere with the operation of instances still in DS5-compatible-mode. |
| DS6-mode | A Directory Server 6.2 instance in DS6-mode uses only new policy attributes in computing password policy decisions. Any old password policy attributes remaining in an entry are ignored. |

The compatibility mode is set using the `dsconf` command as follows:

```
$ dsconf pwd-compat new-mode
```

The *new-mode* action takes one of the following values:

to-DS6-migration-mode	Change to DS6-migration-mode from DS5-compatible-mode. Once the change is made, only DS6-migration-mode and DS6-mode are available.
to-DS6-mode	Change to DS6-mode from DS6-migration-mode. Once the change is made, only DS6-mode is available.

The server state can move only towards stricter compliance with the new password policy specifications. Compatibility with the old password policy will not be supported indefinitely. You should therefore migrate to the new password policy as soon as is feasible for your deployment.

Guidelines for Choosing a Compatibility Mode

The `pwd-compat-mode` setting affects the internal server operation and is largely isolated from the password policy behavior seen by an LDAP client. In particular, the `pwd-compat-mode` setting does not affect the range of server responses to an LDAP client authentication (bind).

Note – The configuration and operational attributes used to implement the password policy depend on the `pwd-compat-mode` setting. Therefore, an LDAP application that accesses the old (Directory Server 5) attributes will need to be modified prior to advancing the `pwd-compat-mode` beyond the initial `DS5-compatible-mode`.

Note – `DS5-compatible-mode` is the default setting. If you upgrade an existing server to Directory Server 6.2 or if you create a new Directory Server 6.2 instance, the compatibility state is set to `DS5-compatible-mode`.

This section provides details about the compatibility mode appropriate to your Directory Server deployment.

New Directory Server 6.2 Deployment

If you install a standalone Directory Server instance or are deploying a new replicated topology, set the compatibility mode to `DS6-mode` to immediately take advantage of the features available in the new password policy implementation. Since a new Directory Server 6.2 instance is

created with the compatibility mode set to `DS5-compatible-mode`, you will need to remember to advance the instance to `DS6-mode` before installing it into a replicated topology whose instances are already set to `DS6-mode`.

Migrating a Deployment from Directory Server 5 to Directory Server 6.2

If you are migrating an existing replicated topology, as long as at least one Directory Server 5 instance remains in the replication topology, all of the Directory Server 6.2 instances must be set to `DS5-compatible-mode`.

Once a replicated topology has been completely migrated from Directory Server 5 to Directory Server 6.2 (in `DS5-compatible-mode`), you can consider advancing from maintaining compatibility with the old password policy to using the new password policy exclusively. Moving from `DS5-compatible-mode` to `DS6-mode` occurs in two phases, which includes an intermediate stage in `DS6-migration-mode`. First, the Directory Server 6.2 instances must be left in `DS5-compatible-mode` for an entire password expiration cycle so that the user entries are populated with the new `pwdChangedTime` attribute. Any applications that depend on the old password policy attributes must also be migrated to the new attributes while the Directory Server 6.2 instances are in `DS5-compatible-mode`, since the old attributes are no longer available in `DS6-migration-mode`. At this point, the instances comprising the replicated topology can be advanced to `DS6-migration-mode`.

The second phase consists of running all instances of the replicated topology in the intermediate `DS6-migration-mode` to clean out the old operational attributes in the entries. This cleanup must occur before advancing from `DS6-migration-mode` to `DS6-mode`. Otherwise, the stale attributes will remain in the entries. To mitigate the overhead of cleaning the old password policy operational attributes, the Directory Server 6.2 instance only removes the attributes in conjunction with a password modify. Thus a simple approach to the cleanup, assuming the password expiration feature is enabled, is to leave the Directory Server 6.2 instances in `DS6-migration-mode` for an entire password expiration cycle. Finally, once the old password policy attributes have been cleaned from the entries, the instances can be moved to `DS6-mode`. Remember that the new Directory Server 6.2 instance is created set to `DS5-compatible-mode`. You will need to remember to advance the instance to `DS6-mode` before installing it into a replicated topology whose instances are already at `DS6-mode`.

The following table shows the allowed combinations of Directory Server versions and password policy compatibility modes. Note that at most two variations are allowed in a replicated topology at any time. For example, if a topology contains a Directory Server 6.2 instance in `DS5-compatible-mode` and one in `DS6-migration-mode`, then those are the only two variants allowed: no Directory Server 5 instances or Directory Server 6.2 instances in `DS6-mode` are allowed.

TABLE 5-4 Directory Server Password Policy Mode Interoperability

	Directory Server 5	Directory Server 6.2 in DS5-compatible-mode	Directory Server 6.2 in DS6-migration-mode	Directory Server 6.2 in DS6-mode
Directory Server 5	X	X		
Directory Server 6.2 in DS5-compatible-mode	X	X	X	
Directory Server 6.2 in DS6-migration-mode		X	X	X
Directory Server 6.2 in DS6-mode			X	X

Changes to Administrative Password Reset Classification

Password policy features such as `must-change-on-reset` (`pwd-must-change-enabled`) and administrative bypass of password quality checks (`pwd-root-dn-bypass-enabled`) depend on classifying the modification of the `userPassword` attribute as either a self-change or an administrative reset.

In Directory Server 5, by default, only the Directory Manager can perform an administrative reset of a user's password. Any other password change is considered as a self-change. Directory Server 5.2p4 introduced the password policy configuration attribute `passwordNonRootMayResetUserpwd` that, when enabled, limits the `userPassword` modify operations that are considered as a self-change to the following two cases:

1. A user is authenticated and changing the password of his or her own account.
2. An administrator changes the password, but the LDAP Proxied Authorization Control (<http://www.ietf.org/rfc/rfc4370.txt>) is set for the `userPassword` modify operation, and the proxied user DN is the target of the modify operation.

Any other password change is considered as an administrative reset. This feature eliminates the requirement of using Directory Manager for routine password administration, while the simple `other-than-self` (password change made by any other user but not by self) test avoids the complexity of a separate scheme to identify administrative users.

Directory Server 6.x evaluates password changes similar to Directory Server 5.x with `passwordNonRootMayResetUserPassword` enabled. That is, Directory Server 6.x considers a password change as an administrative reset except for a user changing his or her own password, or when the proxied authorization control is used. Even though the `passwordNonRootMayResetUserpwd` attribute can be present in a Directory Server 6.x password policy configuration entry when the instance is in Directory Server 5.x compatible mode, the attribute can not be modified and the feature is always enabled.

If your Directory Server 5 based LDAP application uses an administrative account other than Directory Manager to change a password on behalf of a user (that is, the change should be a self-change), when the application is used with Directory Server 6.x, the change will be considered as an administrative reset. You can restore the original behavior by using the LDAP Proxied Authorization Control (<http://www.ietf.org/rfc/rfc4370.txt>) with the userPassword modify operation. The proxied authorization control handles the operation as if it is invoked by the proxied user. The control is available in the LDAP C SDK (http://wiki.mozilla.org/LDAP_C_SDK) and LDAP SDK for Java (<http://www.mozilla.org/directory/javasdk.html>), and the `ldapmodify` command included with DSRK 6. Invoke the proxied authorization control using the `ldapmodify` command as follows:

```
$ ldapmodify -D <administrative-user-DN> -Y <proxied-user-DN>
```

Note – The `ldapmodify` commands from other products might use a different flag, or might not support the proxied authorization control at all.

Changes to Plug-Ins

This section lists the new and deprecated plug-ins in Directory Server 6.2. The section also describes what you need to do if you have custom plug-ins created with the old plug-in API.

New Plug-Ins in Directory Server 6.2

The following plug-ins have been added in Directory Server 6.2:

```
cn=example,cn=ldbm database,cn=plugins,cn=config
cn=gle,cn=plugins,cn=config
cn=MemberOf Plugin,cn=plugins,cn=config
cn=Monitoring Plugin,cn=plugins,cn=config
cn=ObjectDeletionMatch,cn=plugins,cn=config
cn=pswsync,cn=plugins,cn=config
cn=Replication Repair,cn=plugins,cn=config
cn=RMCE,cn>Password Storage Schemes,cn=plugins,cn=config
cn=Strong Password Check,cn=plugins,cn=config
```

For information about these plug-ins see the `plugin(5dsconf)` man page.

Plug-Ins Deprecated in Directory Server 6.2

The following plug-ins have been deprecated in Directory Server 6.2:

```

cn=aci,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=cn,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=encrypted attributes,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=entrydn,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=givenName,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=mail,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=mailHost,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=member,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=monitor,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=nsCalXItemId,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=nscpEntryDN,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=nsRoleDN,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=nsUniqueId,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=nswcalCALID,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=objectclass,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=owner,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=parentid,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=pipstatus,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=pipuid,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=seeAlso,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=sn,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=uid,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=uniquemember,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
cn=userRoot,cn=ldbm database,cn=plugins,cn=config

```

Changes to the Plug-In API

If you have developed your own custom plug-ins, you need to recompile these to work with Directory Server 6.2. For a complete list of the changes made to the plug-in API, see Chapter 2, “Changes to the Plug-In API Since Directory Server 5.2,” in *Sun Java System Directory Server Enterprise Edition 6.2 Developer's Guide*.

Changes to the Installed Product Layout

This section summarizes the changes to the installed product layout from Directory Server 5.2. Several files and utilities have been deprecated since Directory Server 5.2, as described in the following sections.

Administration Utilities Previously Under *ServerRoot*

In Directory Server 6.2 the Administration Server is no longer used to manage server instances.

The following system administration utilities previously located under *ServerRoot* have therefore been deprecated:

- `restart-admin`
- `start-admin`
- `startconsole`
- `stop-admin`
- `uninstall`

Binaries Previously Under *ServerRoot/bin*

The following utilities under *ServerRoot/bin* have been deprecated:

- *ServerRoot/bin/admin/admconfig*
- *ServerRoot/bin/https/bin/ns-httpd*
- *ServerRoot/bin/https/bin/uxwdog*
- *ServerRoot/bin/slapd/server/ns-ldapagt*

On Solaris SPARC, the `ns-slapd` daemon is located in *install-path/ds6/bin/lib/sparcvSolaris-Version*. On platforms other than Solaris SPARC, the `ns-slapd` daemon is located in *install-path/ds6/bin/lib*.

Libraries and Plug-Ins Previously Under *ServerRoot/lib*

Product libraries and plug-ins in Directory Server 5.2 were located under *ServerRoot/lib*. In Directory Server 6.2, on Solaris SPARC, these libraries and plug-ins are located in *install-path/ds6/lib/sparcvSolaris-Version*. On platforms other than Solaris SPARC, they are located directly under *install-path/ds6/lib*.

Online Help Previously Under *ServerRoot/manual*

Console online help files were previously located under *ServerRoot/manual*. The console online help files for Directory Server 6.2 are located under *opt/SUNWdsee/ds6/dccapp/html*.

Plug-Ins Previously Under *ServerRoot/plugins*

The following tables describes the new location of sample server plug-ins, and header files for plug-in development.

TABLE 5-5 Support for Plug-Ins

Directory Server 5.2 Plug-In Directory	Directory Server 6.2 Plug-In Directory	Remarks
<i>ServerRoot/plugins/slapd/slapd/examples</i>	<i>install-path/ds6/examples</i>	Sample plug-ins
<i>ServerRoot/plugins/slapd/slapd/include</i>	<i>install-path/ds6/include</i>	Plug-in header files

SNMP support is no longer handled within Directory Server. SNMP monitoring is now handled by the Java Enterprise System Management Framework (Java ES MF). All plug-ins and binaries related to SNMP have therefore been deprecated within Directory Server.

These plug-ins include the following:

- *ServerRoot/plugins/snmp/magt/magt*
- *ServerRoot/plugins/snmp/mibs/*
- *ServerRoot/plugins/snmp/sagt/sagt*

For information about enabling monitoring Java ES MF monitoring, see “Enabling Java ES MF Monitoring” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

Utilities Previously Under *ServerRoot/shared/bin*

The following tables describes the new location of the administrative tools previously under *ServerRoot/shared/bin*. Note that as a result of the change to the administrative framework, some of these tools have been deprecated.

TABLE 5-6 Tools Previously Under *ServerRoot/shared/bin*

5.2 File	6.0 File	Purpose
<i>ServerRoot/shared/bin/admin_ip.pl</i>	Deprecated	Change IP address
<i>ServerRoot/shared/bin/entrycmp</i>	<i>install-path/ds6/bin/entrycmp</i>	Compare entries for replication
<i>ServerRoot/shared/bin/fildif</i>	<i>install-path/ds6/bin/fildif</i>	Dump filtered LDIF
<i>ServerRoot/shared/bin/insync</i>	<i>install-path/ds6/bin/insync</i>	Check replication synchronization
<i>ServerRoot/shared/bin/ldapcompare</i>	<i>/usr/sfw/bin/ldapcompare</i>	Compare attribute value In Directory Server 6.2 you must install the SUN-LDAPCSDK-TOOLS package to get this utility

TABLE 5-6 Tools Previously Under *ServerRoot*/shared/bin (Continued)

5.2 File	6.0 File	Purpose
<i>ServerRoot</i> /shared/bin/ldapdelete	<i>/usr/sfw/bin/ldapdelete</i>	Delete directory entry In Directory Server 6.2 you must install the SUN-LDAPCSDK-TOOLS package to get this utility
<i>ServerRoot</i> /shared/bin/ldapmodify	<i>/usr/sfw/bin/ldapmodify</i>	Modify directory entry In Directory Server 6.2 you must install the SUN-LDAPCSDK-TOOLS package to get this utility
<i>ServerRoot</i> /shared/bin/ldapsearch	<i>/usr/sfw/bin/ldapsearch</i>	Find directory entries In Directory Server 6.2 you must install the SUN-LDAPCSDK-TOOLS package to get this utility
<i>ServerRoot</i> /shared/bin/modutil	Deprecated	Manage PKCS #11 modules
<i>ServerRoot</i> /shared/bin/uconv	Deprecated	Convert from ISO to UTF-8
<i>ServerRoot</i> /shared/bin/repldisc	<i>install-path/ds6/bin/repldisc</i>	Discover replication topology

Certificate and Key Files

The following table shows the new locations of the certificate and key files in Directory Server 6.2.

TABLE 5-7 Location of Certificate and Key Files

5.2 File	6.0 File	Remarks
<i>ServerRoot</i> /shared/config/certmap.conf	<i>instance-path/alias/certmap.conf</i>	Configuration file for mapping certificates to directory entries
<i>ServerRoot</i> /alias/cert8.db	<i>instance-path/alias/cert8.db</i>	Trusted certificate database file
<i>ServerRoot</i> /alias/key3.db	<i>instance-path/alias/key3.db</i>	Database file containing client keys
<i>ServerRoot</i> /alias/secmod.db	<i>instance-path/alias/secmod.db</i>	Database file containing security modules such as PKCS#11

Silent Installation and Uninstallation Templates

In Directory Server 5.2, the *ServerRoot*/setup5 directory contained sample templates for silent installation and uninstallation. Silent installation and uninstallation are no longer needed for Directory Server 6.2 and these files have therefore been deprecated.

Server Instance Scripts Previously Under *ServerRoot/slapd-ServerID*

The command-line administration scripts previously under *ServerRoot/slapd-ServerID* have been replaced in the new administration framework and deprecated. These commands and their Directory Server 6.2 equivalents are described in [“Command Line Changes” on page 69](#).

Server Instance Subdirectories

The following table describes the new locations for the configuration, log and backup data previously located under *ServerRoot/slapd-instance-name*

TABLE 5-8 Instance-Specific Subdirectories

Version 5 Directory	Version 6 Directory	Remarks
<i>ServerRoot/slapd-ServerID/bak</i>	<i>instance-path/bak</i>	Directory instance database backup
<i>ServerRoot/slapd-ServerID/confbak</i>	Deprecated	Administration Server configuration backup
<i>ServerRoot/slapd-ServerID/conf_bk</i>	<i>instance-path/conf_bk</i>	Directory instance configuration backup
<i>ServerRoot/slapd-ServerID/config</i>	<i>instance-path/config</i>	Directory instance configuration
<i>ServerRoot/slapd-ServerID/config/schema</i>	<i>instance-path/config/schema</i>	Directory instance schema
<i>ServerRoot/slapd-ServerID/db</i>	<i>instance-path/db</i>	Directory instance databases
<i>ServerRoot/slapd-ServerID/ldif</i>	<i>instance-path/ds6/bin/ldif</i>	Sample LDIF files
<i>ServerRoot/slapd-ServerID/locks</i>	<i>instance-path/locks</i>	Run time process locks
<i>ServerRoot/slapd-ServerID/logs</i>	<i>instance-path/logs</i>	Server instance log files
<i>ServerRoot/slapd-ServerID/tmp</i>	<i>instance-path/tmp</i>	Run time temporary files

Migrating Directory Proxy Server

There is no automatic migration path to move from a Directory Proxy Server 5.x version to Directory Proxy Server 6.2. Directory Proxy Server 6.2 provides much more functionality than the 5.x version. While a one to one mapping of configuration information is therefore not possible in most instances, it is possible to configure Directory Proxy Server 6.2 to behave like a version 5 server for compatibility.

This chapter describes how the configuration properties in Directory Proxy Server 6.2 can be used to simulate a version 5 configuration.

The chapter covers the following topics:

- “Mapping the Global Configuration” on page 85
- “Mapping the Connection Pool Configuration” on page 89
- “Mapping the Groups Configuration” on page 90
- “Mapping the Properties Configuration” on page 99
- “Mapping the Events Configuration” on page 105
- “Mapping the Actions Configuration” on page 106
- “Configuring Directory Proxy Server 6.2 as a Simple Connection-Based Router” on page 106

Mapping the Global Configuration

Before you change the Directory Proxy Server 6.2 configuration, back up the configuration by using the `dpadm backup` command. For more information, see `dpadm(1M)`.

You can configure Directory Proxy Server 6.2 by using the Directory Service Control Center (DSCC) or the `dpconf` command-line utility. For more information, see `dpconf(1M)`.

Directory Proxy Server 6.2 configuration can be retrieved as a set of properties. For example, information about the port is returned in the `listen-port` property. This section describes how to map the version 5 global configuration attributes to the corresponding properties in Directory Proxy Server 6.2, where applicable. Not all functionality can be mapped directly.

The global Directory Proxy Server 5 configuration is specified by two object classes:

- **ids-proxy-sch-LDAPProxy.** Contains the name of the Directory Proxy Server server and the DN of the global configuration object.
- **ids-proxy-sch-GlobalConfiguration.** Contains various global configuration attributes.

Because of the way in which Directory Proxy Server 6.2 is configured, Directory Proxy Server 6.2 has no equivalent for the `ids-proxy-sch-LDAPProxy` object class or its attributes.

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Config-Name=name,ou=global,ou=pd2,ou=iDAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ids-proxy-con-Config-Name=user-defined-name,ou=system,ou=dar-config,o=netscaperoot`.

The functionality of the `ids-proxy-sch-GlobalConfiguration` is provided as properties of various elements in Directory Proxy Server 6.2. The following table maps the attributes of the `ids-proxy-sch-GlobalConfiguration` object class to the corresponding properties in Directory Proxy Server 6.2.

TABLE 6-1 Mapping of Version 5 Global Configuration Attributes to 6.0 Properties

Directory Proxy Server 5 Attribute	Directory Proxy Server 6.2 Property
<code>ids-proxy-con-Config-Name</code>	No equivalent
	<p>Directory Proxy Server 6.2 has two <i>listeners</i>, a non-secure listener and a secure listener. The version 5 listen configuration attributes can be mapped to the following four listener properties. To configure listener properties, use the <code>dpconf</code> command as follows:</p> <pre>\$ dpconf set-ldap-listener-prop PROPERTY</pre> <pre>\$ dpconf set-ldaps-listener-prop PROPERTY</pre> <p>For more information, see “Configuring Listeners Between Clients and Directory Proxy Server” in <i>Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide</i>.</p>
<code>ids-proxy-con-listen-port</code>	<code>listen-port</code>
<code>ids-proxy-con-listen-host</code>	<code>listen-address</code>
<code>ids-proxy-con-listen-backlog</code>	<code>max-connection-queue-size</code>
<code>ids-proxy-con-ldaps-port</code>	<code>listen-port</code> (property of the <code>ldaps-listener</code>)

TABLE 6-1 Mapping of Version 5 Global Configuration Attributes to 6.0 Properties (Continued)

Directory Proxy Server 5 Attribute	Directory Proxy Server 6.2 Property
ids-proxy-con-max-conns	<p>This attribute can be mapped to the <code>max-client-connections</code> property of a connection handler resource limit. To configure this property, use the <code>dpconf</code> command as follows:</p> <pre>\$ dpconf set-resource-limit-policy-prop POLICY-NAME max-client-connections:VALUE</pre> <p>For more information, see “Creating and Configuring a Resource Limits Policy” in <i>Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide</i>.</p>
ids-proxy-con-userid	<p>This attribute can be mapped to the user and group names specified when an instance is created by using the following command:</p> <pre>\$ dpadm create [-u NAME -g NAME] INSTANCE-PATH</pre> <p>For more information, see “Creating and Deleting a Directory Proxy Server Instance” in <i>Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide</i>.</p>
ids-proxy-con-working-dir	<p>This attribute can be mapped to the <code>INSTANCE-PATH</code> specified when an instance is created by using the following command:</p> <pre>\$ dpadm create INSTANCE-PATH</pre> <p>For more information, see “Creating and Deleting a Directory Proxy Server Instance” in <i>Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide</i>.</p>
ids-proxy-con-include-logproperties	<p>Not equivalent. For information on configuring logging in Directory Proxy Server 6.2, see Chapter 27, “Directory Proxy Server Logging,” in <i>Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide</i>.</p>

Mapping the Global Security Configuration

In Directory Proxy Server 5, security is configured by using attributes of the global configuration object. In Directory Proxy Server 6.2, you can configure security when you create the server instance by using the `dpadm` command. For more information, see Chapter 19, “Directory Proxy Server Certificates,” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Config-Name=name,ou=global,ou=pd2,ou=iDAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ids-proxy-con-Config-Name=user-defined-name,ou=system,ou=dar-config,o=netscape.root`.

The following table maps the version 5 security attributes to the corresponding properties in Directory Proxy Server 6.

TABLE 6-2 Mapping of Security Configuration

Directory Proxy Server 5 Attribute	Directory Proxy Server 6.2 Property
ids-proxy-con-ssl-key	ssl-key-pin
ids-proxy-con-ssl-cert	ssl-certificate-directory ssl-server-cert-alias
ids-proxy-con-send-cert-as-client This attribute enables the proxy server to send its certificate to the LDAP server to allow the LDAP server to authenticate the proxy server as an SSL client.	ssl-client-cert-alias This property enables the proxy server to send a different certificate to the LDAP server, depending on whether it is acting as an SSL Server or an SSL Client.
ids-proxy-con-server-ssl-version ids-proxy-con-client-ssl-version	No equivalent
ids-proxy-con-ssl-cert-required	This feature can be achieved by setting the following server property: \$ dpconf set-server-prop allow-cert-based-auth:require
ids-proxy-con-ssl-cafile	No equivalent

Managing Certificates

Directory Proxy Server 5, certificates were managed by using the `cert req` utility, or by using the console. In Directory Proxy Server 6.2, certificates are managed by using the `dpadm` command, or by using the DSCC.

Certificates must be installed on each individual data source in Directory Proxy Server 6.2.

For information about managing certificates in Directory Proxy Server 6.2, see Chapter 19, “Directory Proxy Server Certificates,” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

Access Control on the Proxy Configuration

In Directory Proxy Server 5, access control on the proxy configuration is managed by ACIs in the configuration directory server. In Directory Proxy Server 6.2, access to the configuration file is restricted to the person who created the proxy instance, or to the proxy manager if the configuration is accessed through Directory Proxy Server. Editing the configuration file directly is not supported.

Mapping the Connection Pool Configuration

Directory Proxy Server 5 can be configured to reuse existing connections to the backend LDAP servers. This can provide a significant performance gain if the backend servers are on a Wide Area Network (WAN). In Directory Proxy Server 6.2, this functionality is provided with connection pools that are configured in the backend server itself. For more information, see Chapter 20, “LDAP Data Sources and Data Source Pools,” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Config-Name=name,ou=global,ou=pd2,ou=idAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ids-proxy-con-Config-Name=user-defined-name,ou=system,ou=dar-config,o=netscaperoot`.

The following table provides a mapping between Directory Proxy Server 5 connection configuration attributes and the corresponding Directory Proxy Server 6.2 properties.

TABLE 6-3 Mapping of Connection Pool Attributes

Directory Proxy Server 5 Attribute	Directory Proxy Server 6.2 Property
<code>ids-proxy-con-connection-pool</code>	No equivalent
<code>ids-proxy-con-connection-pool-interval</code>	<p>The connection pool grows automatically to a configured maximum. The maximum is configured by setting the following properties of an LDAP data source:</p> <p><code>num-bind-init</code> <code>num-bin-incr</code> <code>num-bind-limit</code> <code>num-read-init</code> <code>num-read-incr</code> <code>num-read-limit</code> <code>num-write-init</code> <code>num-write-incr</code> <code>num-write-limit</code></p> <p>For information about setting LDAP data source properties, see “To Configure an LDAP Data Source” in <i>Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide</i>.</p>
<code>ids-proxy-con-connection-pool-timeout</code>	<code>backendMaxReadWaitTimeInMilliSec</code>

Mapping the Groups Configuration

Directory Proxy Server 5 uses groups to define how client connections are identified and what restrictions are placed on the client connections. In Directory Proxy Server 6.2, this functionality is achieved using connection handlers, data views and listeners.

Connection handlers, data views and listeners can be configured by using the Directory Service Control Center or by using the `dpconf` command. For more information, see Chapter 25, “Connections Between Clients and Directory Proxy Server,” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide* and Chapter 22, “Directory Proxy Server Distribution,” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

Mapping the Group Object

In Directory Proxy Server 5, a group is defined by setting the attributes of the `ids-proxy-sch-Group` object class. Certain attributes of this object class can be mapped to Directory Proxy Server 6.2 connection handler properties. For a list of all the connection-handler properties, run the following command:

```
$ dpconf help-properties | grep connection-handler
```

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=name`, `ou=groups`, `ou=pd2`, `ou=iDAR`, `o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups`, `cn=user-defined-name`, `ou=dar-config`, `o=NetscapeRoot`.

The following table maps version 5 group attributes to the corresponding connection handler properties.

TABLE 6-4 Mapping Between Version 5 Group Attributes and Version 6 Connection Handler Properties

Directory Proxy Server 5 Group Attribute	Directory Proxy Server 6.2 Connection Handler Property
ids-proxy-con-Name	cn
ids-proxy-con-Priority	priority
ids-proxy-sch-Enable	is-enabled
ids-proxy-sch-belongs-to	No equivalent
ids-proxy-con-permit-auth-none:TRUE	allowed-auth-methods:anonymous
ids-proxy-con-permit-auth-sasl:TRUE	allowed-auth-methods:sasl
ids-proxy-con-permit-auth-simple:TRUE	allowed-auth-methods:simple

Mapping the Network Group Object

Directory Proxy Server 5 groups are configured by setting the attributes of the `ids-proxy-sch-NetworkGroup` object class. These attributes can be mapped to properties of Directory Proxy Server 6.2 connection handlers, data sources and listeners. For a list of all the properties related to these objects, run the `dpconf help-properties` command, and search for the object. For example, to locate all the properties of a connection handler, run the following command:

```
$ dpconf help-properties | grep connection-handler
```

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=group-name, ou=groups, ou=pd2, ou=idAR, o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups, cn=user-defined-name, ou=dar-config, o=NetscapeRoot`.

The following table maps Directory Proxy Server 5 network group attributes to the corresponding Directory Proxy Server 6.2 properties and describes how to set these properties by using the command line.

TABLE 6-5 Mapping Between Version 5 Network Group Attributes and 6.2 Properties

Directory Proxy Server 5 Network Group Attribute	Directory Proxy Server 6.2 Property
<code>ids-proxy-con-Client</code>	<code>domain-name-filters</code> and <code>ip-address-filters</code> properties of a connection handler
<code>ids-proxy-con-include-property</code>	No equivalent
<code>ids-proxy-con-include-rule</code>	No equivalent
<code>ids-proxy-con-ssl-policy:ssl_required</code>	Set this as a connection handler property by using the following command: \$ <code>dpconf set-connection-handler-prop CONNECTION-HANDLER-NAME is-ssl-mandatory:true</code>
<code>ids-proxy-con-ssl-policy:ssl_optional</code>	Set this as an LDAP data source property by using the following command: \$ <code>dpconf set-ldap-data-source-prop ds1 ssl-policy:client</code>
<code>ids-proxy-con-ssl-policy:ssl_unavailable</code>	Set this as a connection handler property by using the following command: \$ <code>dpconf set-connection-handler-prop CONNECTION-HANDLER-NAME is-ssl-mandatory:false</code>

TABLE 6-5 Mapping Between Version 5 Network Group Attributes and 6.2 Properties (Continued)

Directory Proxy Server 5 Network Group Attribute	Directory Proxy Server 6.2 Property
ids-proxy-con-tcp-no-delay	Set this as a property for a specific listener port by using the following command: \$ dpconf set-ldap-listener-prop use-tcp-no-delay:true
ids-proxy-con-allow-multi-ldapv2-bind	No equivalent
ids-proxy-con-reverse-dns-lookup	No equivalent
ids-proxy-con-timeout	This functionality exists but with less granularity than in Directory Proxy Server 5. Set this limit as a property for a specific listener port by using the following command: \$ dpconf set-ldap-listener-prop connection-idle-timeout:value

Mapping Bind Forwarding

Directory Proxy Server 5 bind forwarding is used to determine whether to pass a bind request on to an LDAP server or to reject the bind request and close the client's connection. Directory Proxy Server 6.2 forwards either all bind requests or no bind requests. However, by setting the `allowed-auth-methods` connection handler property, successful binds can be classified into connection handlers, according to the authentication criteria. Directory Proxy Server 6.2 can be configured to reject all requests from a specific connection handler, providing the same functionality as Directory Proxy Server 5 bind forwarding.

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=group-name,ou=groups,ou=pd2,ou=iDAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups,cn=user-defined-name,ou=dar-config,o=NetscapeRoot`

The following table maps the Directory Proxy Server 5 bind forwarding attributes to the corresponding Directory Proxy Server 6 connection handler property settings.

TABLE 6-6 Mapping of Directory Proxy Server 5 Bind Forwarding Attributes to Directory Proxy Server 6 Connection Handler Property Settings

Directory Proxy Server 5 Attribute	Directory Proxy Server 6 Property
ids-proxy-con-bind-name	No equivalent
ids-proxy-con-permit-auth-none	<code>allowed-auth-methods:anonymous</code>
ids-proxy-con-permit-auth-simple	<code>allowed-auth-methods:simple</code>

TABLE 6-6 Mapping of Directory Proxy Server 5 Bind Forwarding Attributes to Directory Proxy Server 6 Connection Handler Property Settings (Continued)

Directory Proxy Server 5 Attribute	Directory Proxy Server 6 Property
ids-proxy-con-permit-auth-sasl	allowed-auth-methods:sasl

Mapping Operation Forwarding

Operation forwarding determines how Directory Proxy Server 5 handles requests after a successful bind. In Directory Proxy Server 6.2, this functionality is provided by setting the properties of a request filtering policy. For information on configuring a request filtering policy, see “Creating and Configuring Request Filtering Policies and Search Data Hiding Rules” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*. For a list of all the properties of a request filtering policy, run the following command:

```
$ dpconf help-properties | grep request-filtering-policy
```

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=group-name,ou=groups,ou=pd2,ou=iDAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups,cn=user-defined-name,ou=dar-config,o=NetscapeRoot`.

The following table maps the Directory Proxy Server 5 operation forwarding attributes to the corresponding Directory Proxy Server 6 request filtering properties.

TABLE 6-7 Mapping of Directory Proxy Server 5 Operation Forwarding Attributes to Directory Proxy Server 6 Request Filtering Properties

Directory Proxy Server 5 Attribute	Directory Proxy Server 6 Property
ids-proxy-con-permit-op-search	allow-search-operations
ids-proxy-con-permit-op-compare	allow-compare-operations
ids-proxy-con-permit-op-add	allow-add-operations
ids-proxy-con-permit-op-delete	allow-delete-operations
ids-proxy-con-permit-op-modify	allow-modify-operations
ids-proxy-con-permit-op-modrdn	allow-rename-operations
ids-proxy-con-permit-op-extended	allow-extended-operations

Mapping Subtree Hiding

Directory Proxy Server 5 uses the `ids-proxy-con-forbidden-subtree` attribute to specify a subtree of entries to be excluded in any client request. Directory Proxy Server 6.2 provides this functionality with the `allowed-subtrees` and `prohibited-subtrees` properties of a request filtering policy. For information on hiding subtrees in this way, see “Creating and Configuring a Resource Limits Policy” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

If your subtrees are distributed across different backend servers, you can use the `excluded-subtrees` property of a data view to hide subtrees. For more information on hiding subtrees in this way, see “Excluding a Subtree From a Data View” in *Sun Java System Directory Server Enterprise Edition 6.2 Reference* and “To Configure Data Views With Hierarchy and a Distribution Algorithm” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

Mapping Search Request Controls

In Directory Proxy Server 5, search request controls are used to prevent certain kinds of requests from reaching the LDAP server. In Directory Proxy Server 6.2, this functionality is provided by setting properties of a request filtering policy and a resource limits policy.

For information on configuring a request filtering policy, see “Creating and Configuring Request Filtering Policies and Search Data Hiding Rules” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*. For information on configuring a resource limits policy, see “Creating and Configuring a Resource Limits Policy” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*. For a list of all the properties associated with a request filtering policy, or a resource limits policy, run the `dpadm help-properties` command and search for the object. For example, to locate all properties associated with a resource limits policy, run the following command:

```
$ dpconf help-properties | grep resource-limits-policy
```

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=group-name,ou=groups,ou=pd2,ou=idAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups,cn=user-defined-name,ou=dar-config,o=NetscapeRoot`.

The following table maps the Directory Proxy Server 5 search request control attributes to the corresponding Directory Proxy Server 6.2 properties.

TABLE 6-8 Mapping Directory Proxy Server 5 Search Request Control Attributes to Directory Proxy Server 6.2 Properties

Directory Proxy Server 5 Attribute	Directory Proxy Server 6.2 Property
ids-proxy-con-filter-inequality	allow-inequality-search-operations property of the request filtering policy
ids-proxy-con-min-substring-size	minimum-search-filter-substring-length property of the resource limits policy

Mapping Compare Request Controls

In Directory Proxy Server 5, compare request controls are used to prevent certain kinds of search and compare operations from reaching the LDAP server. In Directory Proxy Server 6.2, this functionality is provided by setting properties of a request filtering policy.

For information on configuring a request filtering policy, see “Creating and Configuring Request Filtering Policies and Search Data Hiding Rules” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=group-name,ou=groups,ou=pd2,ou=idAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups,cn=user-defined-name,ou=dar-config,o=NetscapeRoot`.

The following table maps the Directory Proxy Server 5 compare request control attributes to the corresponding Directory Proxy Server 6 properties.

TABLE 6-9 Mapping of Directory Proxy Server 5 Compare Request Control Attributes to Directory Proxy Server 6 Properties

Directory Proxy Server 5 Attribute	Directory Proxy Server 6 Property
ids-proxy-con-forbidden-compare	prohibited-comparable-attrs
ids-proxy-con-permitted-compare	allowed-comparable-attrs

Mapping Attributes Modifying Search Requests

In Directory Proxy Server 5, these attributes are used to modify the search request before it is forwarded to the server. In Directory Proxy Server 6, this functionality is provided by setting properties of a request filtering policy and a resource limits policy.

For information on configuring a request filtering policy, see “Creating and Configuring Request Filtering Policies and Search Data Hiding Rules” in *Sun Java System Directory Server*

Enterprise Edition 6.2 Administration Guide. For information on configuring a resource limits policy, see “Creating and Configuring a Resource Limits Policy” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=group-name,ou=groups,ou=pd2,ou=iDAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups,cn=user-defined-name,ou=dar-config,o=NetscapeRoot`.

The following table maps the Directory Proxy Server 5 search request modifying attributes to the corresponding Directory Proxy Server 6 properties.

TABLE 6-10 Mapping of Directory Proxy Server 5 Search Request Modifying Attributes to Directory Proxy Server 6 Properties

Directory Proxy Server 5 Attribute	Directory Proxy Server 6 Property
ids-proxy-con-minimum-base	allowed-subtrees property of the request filtering policy
ids-proxy-con-max-scope	allowed-search-scopes property of the request filtering policy
ids-proxy-con-max-timelimit	search-time-limit property of the resource limits policy

Mapping Attributes Restricting Search Responses

In Directory Proxy Server 5, these attributes describe restrictions that are applied to search results being returned by the server, before they are forwarded to the client. In Directory Proxy Server 6, this functionality is provided by setting the properties of a resource limits policy and by configuring search data hiding rules.

For information about configuring a resource limits policy, see “Creating and Configuring a Resource Limits Policy” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*. For information about creating search data hiding rules, see “To Create Search Data Hiding Rules” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*. For a list of properties associated with a search data hiding rule, run the following command:

```
$ dpconf help-properties | grep search-data-hiding-rule
```

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=group-name,ou=groups,ou=pd2,ou=iDAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups,cn=user-defined-name,ou=dar-config,o=NetscapeRoot`.

The following table maps the Directory Proxy Server 5 search response restriction attributes to the corresponding Directory Proxy Server 6.2 properties.

TABLE 6-11 Mapping of Directory Proxy Server 5 Search Response Restriction Attributes to Directory Proxy Server 6.2 Properties

Directory Proxy Server 5 Attributes	Directory Proxy Server 6.2 Properties
<code>ids-proxy-con-max-result-size</code>	<code>search-size-limit</code> property of the resource limits policy
<code>ids-proxy-con-forbidden-return</code>	To hide a subset of attributes: <code>rule-action:hide-attributes</code> <code>attributes:attribute-name</code> To hide an entire entry: <code>rule-action:hide-entry</code>
<code>ids-proxy-con-permitted-return</code>	<code>rule-action:show-attributes</code> <code>attributes:attribute-name</code>
<code>ids-proxy-con-search-reference</code>	No direct equivalent. Search continuation references are governed by the <code>referral-policy</code> property of the resource limits policy

Mapping the Referral Configuration Attributes

In Directory Proxy Server 5, these attributes determine what Directory Proxy Server should do with referrals. In Directory Proxy Server 6.2, this functionality is provided by setting properties of a resource limits policy.

For information on configuring a resource limits policy, see “Creating and Configuring a Resource Limits Policy” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=group-name`, `ou=groups`, `ou=pd2`, `ou=iDAR`, `o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups`, `cn=user-defined-name`, `ou=dar-config`, `o=NetscapeRoot`.

The following table maps the Directory Proxy Server 5 referral configuration attributes to the corresponding Directory Proxy Server 6 resource limits properties.

TABLE 6-12 Mapping of Directory Proxy Server 5 Referral Configuration Attributes to Directory Proxy Server 6 resource limits Properties

Directory Proxy Server 5 Attribute	Directory Proxy Server 6 Property
ids-proxy-con-reference	referral-policy
ids-proxy-con-referral-ssl-policy	referral-policy
ids-proxy-con-referral-bind-policy	referral-bind-policy
ids-proxy-con-max-refcount	referral-hop-limit

Mapping the Server Load Configuration

In Directory Proxy Server 5, these attributes are used to control the number of simultaneous operations and total number of operations a client can request on one connection. In Directory Proxy Server 6, this functionality is provided by setting properties of a resource limits policy.

For information on configuring a resource limits policy, see “Creating and Configuring a Resource Limits Policy” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=group-name,ou=groups,ou=pd2,ou=iDAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups,cn=user-defined-name,ou=dar-config,o=NetscapeRoot`.

The following table maps the Directory Proxy Server 5 server load configuration attributes to the corresponding Directory Proxy Server 6.2 resource limits properties.

TABLE 6-13 Mapping of Directory Proxy Server 5 Server Load Configuration Attributes to Directory Proxy Server 6.2 Resource Limits Properties

Directory Proxy Server 5 Attribute	Directory Proxy Server 6.2 Property
ids-proxy-con-max-simultaneous-operations-per-connection	max-simultaneous-operations-per-connection
ids-proxy-con-operations-per-connection	max-total-operations-per-connection
ids-proxy-con-max-conns	max-connections
ids-proxy-con-max-simultaneous-conns-from-ip	max-client-connections

Mapping the Properties Configuration

The Directory Proxy Server 5 property objects enable you to specify specialized restrictions that LDAP clients must follow. Most of the functionality of property objects is available in Directory Proxy Server 6, although it is supplied by various elements of the new architecture. The following sections describe how to map the Directory Proxy Server 5 property objects to the corresponding 6.0 functionality.

Attribute Renaming Property

In Directory Proxy Server 5, attribute renaming is defined by the `ids-proxy-sch-RenameAttribute` object class. This object uses the `ids-proxy-con-server-attr-name` and `ids-proxy-con-client-attr-name` attributes to specify which attributes must be renamed by Directory Proxy Server.

The attribute renaming functionality is replaced in Directory Proxy Server 6 by the `attr-name-mappings` property of an LDAP data source. This property is multi-valued, and takes values of the form *client-attribute-name#server-attribute-name*. In a client request, Directory Proxy Server renames the `client-attribute-name` to the `server-attribute-name`. In a response, Directory Proxy Server renames the `server-attribute-name` to the `client-attribute-name`.

To configure this property, use the following command:

```
$ dpconf set-ldap-data-source-prop data-source-name \
  attr-name-mappings:client-attribute-name#server-attribute-name
```

Forbidden Entry Property

In Directory Proxy Server 5, the `ids-proxy-sch-ForbiddenEntryProperty` object is used to specify a list of entries or attributes that are hidden from client applications. In Directory Proxy Server 6.2 this functionality is achieved by creating a `search-data-hiding-rule` for a request filtering policy.

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=group-name,ou=groups,ou=pd2,ou=idAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups,cn=user-defined-name,ou=dar-config,o=NetscapeRoot`.

The following table maps the attributes of the `ids-proxy-sch-ForbiddenEntryProperty` object to the corresponding properties of a search data hiding rule in Directory Proxy Server 6.2. For information about creating search data hiding rules, see “To Create Search Data Hiding Rules” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

TABLE 6-14 Mapping of Directory Proxy Server 5 Server Load Configuration Attributes to Directory Proxy Server 6 Resource Limits Properties

Directory Proxy Server 5 Attribute	Directory Proxy Server 6 Property
ids-proxy-con-dn-exact	target-dns
ids-proxy-con-dn-regexp	target-dn-regular-expressions
ids-proxy-con-ava	target-attr-value-assertions
ids-proxy-con-forbidden-return	To hide a subset of attributes: rule-action:hide-attributes attrs:attribute-name To hide an entire entry: rule-action:hide-entry
ids-proxy-con-permitted-return	rule-action:show-attributes attrs:attribute-name

LDAP Server Property

In Directory Proxy Server 5, the `ids-proxy-sch-LDAPServer` property is used to define the backend LDAP servers to which Directory Proxy Server sends requests. In Directory Proxy Server 6.2, this functionality is achieved by using LDAP data sources. You can set properties for LDAP data sources by using the Directory Service Control Center or by using the command line. For more information, see “Creating and Configuring LDAP Data Sources” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=server-name,ou=properties,ou=pd2,ou=iDAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups,cn=user-defined-name,ou=dar-config,o=NetscapeRoot`.

The following table maps the attributes of the `ids-proxy-sch-LDAPServer` object class to the corresponding data source properties in Directory Proxy Server 6.2. Data sources provide additional functionality that was not provided in Directory Proxy Server 5. Not all data source properties are listed here. For a list of all the properties that can be configured for a data source, run the following command:

```
$ dpconf help-properties | grep ldap-data-source
```

TABLE 6-15 Mapping of `ids-proxy-sch-LDAPServer` Attributes to Data Source Properties

Directory Proxy Server 5 Attribute	Directory Proxy Server 6.2 Property
<code>ids-proxy-con-host</code>	<code>ldap-address</code>
<code>ids-proxy-con-port</code>	<code>ldap-port</code>
<code>ids-proxy-con-sport</code>	<code>ldaps-port</code>
<code>ids-proxy-con-supported-version</code>	No equivalent Directory Proxy Server 6.2 supports LDAP v3 backends for both version 2 and version 3 clients. Directory Proxy Server 6.2 supports the proxy authorization control version 1 and version 2.
<code>ids-proxy-con-use-version</code>	No equivalent Directory Proxy Server 6.2 supports LDAP v3 backends for both v2 and v3 clients. Directory Proxy Server 6.2 supports the proxy authorization control version 1 and version 2.
<code>ids-proxy-con-tcp-no-delay</code>	<code>use-tcp-no-delay</code>
<code>ids-proxy-con-link-security-policy</code>	<code>ssl-policy</code>
<code>ids-proxy-con-x509cert-subject</code>	No equivalent. Directory Proxy Server 6.2 does not check the subject of the certificate provided by the backend server.
<code>ids-proxy-con-keepalive-interval</code>	This functionality is achieved by setting the following properties of the LDAP data source: <code>monitoring-bind-timeout</code> <code>monitoring-entry-timeout</code> <code>monitoring-inactivity-timeout</code> <code>monitoring-interval</code> For information about setting LDAP data source properties, see “To Configure an LDAP Data Source” in <i>Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide</i> .

Load Balancing Property

In Directory Proxy Server 5, the `ids-proxy-sch-LoadBalanceProperty` is used to configure load balancing across multiple LDAP servers. Directory Proxy Server 5 supports proportional

load balancing only, that is, each LDAP server is allotted a certain percentage of the total load. The `ids-proxy-sch-LoadBalanceProperty` object class has one attribute, `ids-proxy-con-Server`, whose value has the following syntax:

```
server-name[#percentage]
```

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=load-balance,ou=properties,ou=pd2,ou=iDAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ids-proxy-con-name=load-balancing-1,ou=properties,cn=user-defined-name,ou=dir-config,o=NetSc`

In Directory Proxy Server 6.2, load balancing is configured as a property of a data source pool. A data source pool is essentially a collection of LDAP servers to which Directory Proxy Server can route requests. For information about setting up a data source pool, see “Creating and Configuring LDAP Data Source Pools” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*. For a list of properties associated with a data source pool, run the following command:

```
$ dpconf help-properties | grep ldap-data-source-pool
```

Directory Proxy Server 6.2 supports proportional load balancing but also supports additional load balancing algorithms. To configure proportional load balancing, set the property of the data source pool as follows:

```
$ dpconf set-ldap-data-source-pool-prop data-source-pool-name load-balancing-algorithm:proportional
```

The percentage of load allotted to each server is configured by setting various properties of an attached data source. An attached data source is a data source that has been attached to a specific data source pool. To configure proportional load, set the weight properties of the attached data source for each operation type as follows:

```
$ dpconf set-attached-ldap-data-source-prop data-source-pool-name attached-data-source-name
add-weight:value
bind-weight:value
compare-weight:value
delete-weight:value
modify-dn-weight:value
modify-weight:value
search-weight:value
```

For more information, see “Configuring Load Balancing” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

Monitoring Backend Servers

To monitor the state of its backend LDAP servers, Directory Proxy Server 5 performs an anonymous search operation on the RootDSE of each server every ten seconds. Directory Proxy

Server 6.2 has a number of properties that can be configured to monitor its backend servers. For more information, see “Retrieving Monitored Data About Data Sources” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

Search Size Limit Property

Directory Proxy Server 5 uses the `ids-proxy-sch-SizeLimitProperty` to apply size limits based on the base and scope of search operations. In Directory Proxy Server 6.2, the search size limit can be configured by setting a property of the resource limits policy. A resource limits policy defines the maximum resource that Directory Proxy Server can process for a given connection handler. Use the `dpconf` command to set the search size limit for a resource policy, as follows:

```
$ dpconf set-resource-limits-policy-prop policy-name search-size-limit:number-of-entries
```

Resource limits policies control much more than just search size limit. For information on configuring resource limits policies, see “Creating and Configuring a Resource Limits Policy” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Name=group-name,ou=groups,ou=pd2,ou=iDAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ou=groups,cn=user-defined-name,ou=dar-config,o=NetscapeRoot`.

The following table maps the attributes of a version 5 size limit property to the corresponding properties in Directory Proxy Server 6.2.

TABLE 6-16 Mapping of Version 5 Search Size Limit Attributes to 6.0 Properties

Directory Proxy Server 5 Attribute	Directory Proxy Server 6.2 Property
<code>ids-proxy-con-Size-Limit</code>	<code>search-size-limit</code>
<code>ids-proxy-con-Dn-One</code>	<code>one-level-search-base-dn</code>
<code>ids-proxy-con-Dn-Sub</code>	No equivalent

Log Property

The logging functionality available in Directory Proxy Server 5 differs substantially from the functionality available in Directory Proxy Server 6.2.

In Directory Proxy Server 5, the following logs were maintained:

- **System log.** Includes log records of system events and errors.
- **Audit log.** Includes audit trails for all events and errors.

Directory Proxy Server 6.2 maintains an errors log file, an access log file, and administrative alerts.

The errors log and administrative alerts are equivalent to the version 5 system log. Administrative alerts are events raised by Directory Proxy Server. These events can be sent to the `syslog` daemon or to an administrator through email.

The Directory Proxy Server 6.2 access log is equivalent to the version 5 audit log.

Logs in version 5 were configured by using the `ids-proxy-sch-LogProperty` object class. Logs in Directory Proxy Server 6.2 are configured by setting properties for the access and error log, using the `dpconf` command. For example, to set properties for the access log, use the following command:

```
$ dpconf set-access-log-prop PROPERTY:VALUE
```

Directory Proxy Server 6.2 provides new log features, such as log file rotation, and enables log configuration to be fine tuned. For example, one log level can be set per message category.

In iPlanet Directory Access Router 5.0 (IDAR) log configuration attributes are stored under `ids-proxy-con-Config-Name=name, ou=global, ou=pd2, ou=iDAR, o=services`. In Directory Proxy Server 5.2, log configuration attributes are stored under `ids-proxy-con-Config-Name=user-defined-name, ou=system, ou=dar-config, o=netscaperoot`.

It is not really possible to map the log configuration between Directory Proxy Server 5 and Directory Proxy Server 6.2 because the logging models between these two versions are very different. The Directory Proxy Server 5 log model combines what is logged with where it is logged. In Directory Proxy Server 6.2, the model is cleaner. One set of properties describes what is logged, and a separate set of properties describes where log messages are sent.

The following table lists the attributes of the `ids-proxy-sch-LogProperty` object class and describes at a high level how the corresponding functionality is achieved in Directory Proxy Server 6.2.

TABLE 6-17 Version 5 and Version 6 Log Functionality

Directory Proxy Server 5 Attribute	Purpose	Directory Proxy Server 6.2 Equivalent
<code>ids-proxy-con-log-level</code>	Level of logging	Global log level
<code>ids-proxy-con-stat-level</code>	Kinds of statistics logged	Monitoring data
<code>ids-proxy-con-log-syslog</code>	Syslog facility code	<code>syslog</code> output for administrative alerts No equivalent for error messages
<code>ids-proxy-con-log-file</code>	Path to log file	<code>log-file-name</code> of the <code>error-log</code> object

TABLE 6-17 Version 5 and Version 6 Log Functionality (Continued)

Directory Proxy Server 5 Attribute	Purpose	Directory Proxy Server 6.2 Equivalent
ids-proxy-con-audit-syslog	Syslog facility code for audit log	No equivalent
ids-proxy-con-audit-file	Path to audit log file	log-file-name of the access-log object

Because a one to one mapping of log configuration is not possible between the two versions, you need to understand the new logging model and then configure your new logs accordingly, rather than migrating your old log configuration. For more information, see Chapter 27, “Directory Proxy Server Logging,” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

Mapping the Events Configuration

Directory Proxy Server 5 event objects are used to specify conditions that Directory Proxy Server should evaluate at predetermined states.

Two types of event objects are supported:

- **OnBindSuccess.** Evaluated when a client successfully completes a bind operation.
- **OnSSLEstablished.** Evaluated when a client successfully established an SSL session.

In Directory Proxy Server 6.2, events are implemented as properties of a connection handler. Use the `dpconf` command to set these properties. For example, run the following command to set the authentication methods for the connection handler:

```
$ dpconf set-connection-handler-prop connection-handler-name \
  allowed-auth-methods:anonymous allowed-auth-methods:sasl allowed-auth-methods:simple
```

In iPlanet Directory Access Router 5.0 (IDAR) these configuration attributes are stored under `ids-proxy-con-Config-Name=name,ou=global,ou=pd2,ou=idAR,o=services`. In Directory Proxy Server 5.2, these configuration attributes are stored under `ids-proxy-con-Config-Name=user-defined-name,ou=system,ou=dar-config,o=netscaperoot`.

The following table maps the version 5 event configuration attributes to the corresponding properties in Directory Proxy Server 6.2.

TABLE 6-18 Mapping Between Version 5 Event Attributes and Version 6 Connection Handler Properties

Directory Proxy Server 5 Attribute	Directory Proxy Server 6.2 Property
ids-proxy-sch-OnBindSuccessRule	bind-dn-filters

TABLE 6-18 Mapping Between Version 5 Event Attributes and Version 6 Connection Handler Properties (Continued)

Directory Proxy Server 5 Attribute	Directory Proxy Server 6.2 Property
ids-proxy-con-ssl-required	is-ssl-mandatory
ids-proxy-con-bind-anonymous	allowed-auth-methods:anonymous
ids-proxy-con-bind-simple	allowed-auth-methods:simple
ids-proxy-con-bind-sasl	allowed-auth-methods:sasl

Mapping the Actions Configuration

Directory Proxy Server 5 supports only one action, specified by the `ids-proxy-sch-ChangeGroupAction` object class. This action enables you to configure Directory Proxy Server to change a client from one access group to another based on the evaluation of a rule. The action uses the multi-valued `ids-proxy-con-to-group` attribute to specify the groups to which the client can change.

Directory Proxy Server 6.2 connection handlers provide this functionality. After being classified into a connection handler, a connection can be automatically reclassified into another connection handler. For example, if a client connects anonymously, the connection is allocated to the connection handler configured for anonymous connections. If the client later provides a bind DN on the same connection, the connection can be reallocated to another connection handler.

For information on how to configure this functionality in Directory Proxy Server 6.2, see “Creating, Configuring, and Deleting Connection Handlers” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*.

Configuring Directory Proxy Server 6.2 as a Simple Connection-Based Router

It is possible to configure an instance of Directory Proxy Server 6.2 to behave as a simple connection-based router, with the same functionality as Directory Proxy Server 5.2. To do this, map the configuration attributes described previously and follow the procedure describe in “Configuring Directory Proxy Server as a Connection Based Router” in *Sun Java System Directory Server Enterprise Edition 6.2 Administration Guide*

Migrating Identity Synchronization for Windows

This chapter explains how to migrate your system from Identity Synchronization for Windows version 1.1, and 1.1 SP1, to version 6.0.

In the remainder of this chapter, version 1.1 includes version 1.1 SP1.

Note – When you install Identity Synchronization for Windows version 1.1, Message Queue is also installed on your system. Identity Synchronization for Windows 6.0 does *not* install Message Queue.

For installation and upgrade information about Message Queue, read the installation instructions for Java Enterprise System software at <http://docs.sun.com/coll/1286.2> (<http://docs.sun.com/coll/1286.2>).

This chapter includes the following sections:

- “Migration Overview” on page 108
- “Before You Migrate Identity Synchronization for Windows” on page 108
- “Preparing for Identity Synchronization for Windows Migration” on page 109
- “Migrating Your System” on page 118
- “What to Do if the 1.1 Uninstallation Fails” on page 127
- “Other Migration Scenarios” on page 141
- “Checking the Logs” on page 146

Migration Overview

Migration from Identity Synchronization for Windows version 1.1 to version 6.0 is accomplished in the following major phases:

1. Preparing your Identity Synchronization for Windows 1.1 installation for migration.
2. Uninstalling Identity Synchronization for Windows 1.1.
3. Installing or upgrading dependent products.
4. Installing Identity Synchronization for Windows 6.0 by using the configuration and connector states you backed up.

Note – Install Identity Synchronization for Windows 6.0 on the same platform and architecture where you installed Identity Synchronization for Windows 1.1.

Before You Migrate Identity Synchronization for Windows

Complete the following tasks before you migrate:

- Familiarize yourself with the new features and functionality provided in Identity Synchronization for Windows 6.0.
- Read Chapter 4, “Understanding the Product,” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide* for installation and configuration information that you can use to plan your migration process.
- Document your version 1.1 deployment and configuration. Be sure to note any customizations you have made to the configuration.
- Schedule migration. Because the migration process requires at least four hours, you might want to schedule migration after normal business hours.

If the input password or attribute changes while you are migrating the system, Identity Synchronization for Windows processes these changes as follows:

- **For Active Directory.** Any password changes made on Active Directory during the migration process will be synchronized on demand by the Directory Server Plug-in after the migration process.
- **For Directory Server.** Any password changes made on Directory Server during the migration process will not be synchronized. However, you can identify affected users in the Identity Synchronization for Windows 6.0 logs after completing the migration process. For more information, see [“Checking the Logs” on page 146](#).
- **For Windows NT.** Any password changes made on NT during the migration process will not be synchronized.

However, if you use the `forcepwchg` utility, you can identify affected users and force them to change passwords again. For more information, see [“Forcing Password Changes on Windows NT” on page 118](#).

- All other attribute changes made during the migration process (at any directory source) will be synchronized after the migration process.

Preparing for Identity Synchronization for Windows Migration

Use one or more of the following utilities to migrate from Identity Synchronization for Windows 1.1 to Identity Synchronization for Windows 6.0:

- **export11cnf.** A stand-alone utility that enables you to create an export configuration file from your Identity Synchronization for Windows 1.1 configuration. For more information, see [“Exporting Version 1.1 Configuration” on page 109](#).

The exported XML document contains the directory deployment topology and enough information to configure the Identity Synchronization for Windows 6.0 installation.

- **checktopics.** A utility that checks Message Queue synchronization topics in a 1.1 installation and determines if any undelivered messages remain in the queue.

Updates can remain in Message Queue after you stop 1.1 synchronization. You must verify that no updates exist in the Message Queue before you proceed with the migration. For more information, see [“Checking for Undelivered Messages” on page 116](#).

- **forcepwchg.** A Windows NT tool that enables you to identify users who changed passwords during the migration process and forces them to change passwords again when the version 6.0 system is ready. Password changes made on Windows NT are not captured during the migration process. For more information, see [“Forcing Password Changes on Windows NT” on page 118](#) for detailed information.

Note – These utilities facilitate the migration of Identity Synchronization for Windows version 1.1 to version 6.0. The migration is performed in the same environment where Identity Synchronization for Windows 1.1 is deployed. Consequently, these utilities are available in the Solaris/SPARC and Windows packages only.

You can find the migration utilities in the installation migration directory. No additional installation steps are required.

Exporting Version 1.1 Configuration

You can use the `export11cnf` utility to export an existing version 1.1 configuration file to an XML file and then use the `idsync importcnf` command to import the file into the Identity Synchronization for Windows 6.0 system before installing the connectors.

Tip – While you can update the 1.1 system configuration manually by using the Identity Synchronization for Windows console, we recommend that you use the `export11cnf` utility. If you do not use `export11cnf`, the state of the connectors is not preserved.

Exporting the version 1.1 configuration enables you to:

- Eliminate most of the initial configuration process to be performed from the management Console.
- Guarantee that the connector IDs assigned in version 6.0 match the connector IDs used in version 1.1. This simplifies the task of preserving the existing connector states that can be used directly in the version 6.0 deployment.

Back up the `persist` and `etc` directories, and then restore them later to avoid confusion about the underlying directory structure.

You can find the `export11cnf` utility in the `installation migration` directory. No additional installation steps are necessary.

Using the `export11cnf` Utility

To export an Identity Synchronization for Windows configuration to an XML file, execute `export11cnf` from the `migration` directory as follows:

In a terminal window, type the following:

```
java -jar export11cnf.jar -h hostname
-p port -D bind DN
-w bind password -s rootsuffix
-q configuration password -Z -P cert-db-path
-m secmod-db-path -f filename
```

For example,

```
java -jar export11cnf.jar -D "cn=dirmanager" -w - -q - -s "dc=example,dc=com" -f
exported-configuration
```

The `export11cnf` utility shares the same common arguments as the Identity Synchronization for Windows command-line utilities. For more information, see “Common Arguments to the Idsync Subcommands” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*. The `export11cnf` utility exports the current configuration into the file specified in the argument of the `-f` option.

Inserting Clear-Text Passwords

For security reasons, the `export11cnf` utility does not export clear-text passwords from version 1.1. Instead, the utility inserts empty strings in `cleartextPassword` fields wherever applicable.

For example,

```
<Credentials
  userName="cn=iswservice,cn=users,dc=example,dc=com"
  cleartextPassword=""/>
<!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD -->
```

You must enter a password manually, between double quotes, for every `cleartextPassword` field in the exported configuration file, before you can import the file into Identity Synchronization for Windows. `importcnf` validation prevents you from importing a configuration file with empty password values.

For example,

```
<Credentials
  userName="cn=iswservice,cn=users,dc=example,dc=com"
  cleartextPassword="mySecretPassword"/>
<!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD -->
```

Sample Export Configuration File

In the following sample exported configuration file,

- `ad-host.example.com` refers to the Active Directory domain controller.
- `ds-host.example.com` refers to the host running Directory Server.

EXAMPLE 7-1 Sample Export Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>

<ActiveConfiguration>
  <SunDirectorySource
    parent.attr="DirectorySource"
    onDemandSSLOption="true"
    maxConnections="5"
    displayName="dc=example,dc=com"
    resyncInterval="1000">

    <SynchronizationHost
      hostOrderOfSignificance="1"
      hostname="ds-host.example.com"
      port="389"
      portSSLOption="true"
      securePort="636"/>

    <Credentials
      userName="uid=PSWConnector,
        dc=example,
        dc=com"

    </SynchronizationHost>
  </SyncScopeDefinitionSet
```

EXAMPLE 7-1 Sample Export Configuration File (Continued)

```

        index="0"
        location="ou=people,dc=example,dc=com"
        filter=""
        creationExpression="uid=%uid%,ou=people,dc=example,dc=com"
        sulid="SUL1"/>
</SunDirectorySource>

<ActiveDirectorySource
  parent.attr="DirectorySource"
  displayName="example.com"
  resyncInterval="1000">
  <SynchronizationHost
    hostOrderOfSignificance="1"
    hostname="ad-host.example.com"
    port="389"
    portSSLOption="true"
    securePort="636">
    <Credentials
      userName="cn=Administrator,cn=Users,dc=metaqa,dc=com"
      clearTextPassword=""/>
      <!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD -->
    </SynchronizationHost>
    <SyncScopeDefinitionSet
      index="0"
      location="cn=users,dc=example,dc=com"
      filter=""
      creationExpression="cn=%cn%,cn=users,dc=example,dc=com"
      sulid="SUL1"/>
    </ActiveDirectorySource>

<ActiveDirectoryGlobals
  flowInboundCreates="true"
  flowInboundModifies="true"
  flowOutboundCreates="true"
  flowOutboundModifies="true">
  <TopologyHost
    parent.attr="SchemaLocation"
    hostname="ad-host.example.com"
    port="3268"
    portSSLOption="true"
    securePort="3269">
    <Credentials
      parent.attr="Credentials"
      userName="cn=Administrator,cn=Users,dc=example,dc=com"

```


EXAMPLE 7-1 Sample Export Configuration File (Continued)

```

    cleartextPassword=""/>
    <!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD -->
</TopologyHost>

<TopologyHost
    parent.attr="HostsTopologyConfiguration"
    hostname="ad-host.example.com"
    port="3268"
    portSSLOption="true"
    securePort="3269">
    <Credentials
        parent.attr="Credentials"
        userName="cn=Administrator,cn=Users,dc=example,dc=com"
        cleartextPassword=""/>
    <!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD -->
</TopologyHost>

<AttributeMap>
    <AttributeDescription
        parent.attr="WindowsAttribute"
        name="lockouttime"
        syntax="1.2.840.113556.1.4.906"/>
    <AttributeDescription
        parent.attr="SunAttribute"
        name="pwdaccountlockedtime"
        syntax="1.3.6.1.4.1.1466.115.121.1.24"/>
</AttributeMap>

<AttributeDescription
    parent.attr="SignificantAttribute"
    name="lockouttime"
    syntax="1.2.840.113556.1.4.906"/>
<AttributeDescription
    parent.attr="SignificantAttribute"
    name="samaccountname"
    syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeDescription
    parent.attr="CreationAttribute"
    name="samaccountname"
    syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeMap>
    <AttributeDescription
        parent.attr="WindowsAttribute"
        name="samaccountname"
        syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
    <AttributeDescription

```

EXAMPLE 7-1 Sample Export Configuration File (Continued)

```

        parent.attr="SunAttribute"
        name="uid"
        syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
</AttributeMap>

<AttributeMap>
  <AttributeDescription
    parent.attr="SunAttribute"
    name="sn"
    syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
  <AttributeDescription
    parent.attr="WindowsAttribute"
    name="sn"
    syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
</AttributeMap>

<AttributeDescription
  parent.attr="SignificantAttribute"
  name="sn"
  syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeDescription
  parent.attr="SignificantAttribute"
  name="cn"
  syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeDescription
  parent.attr="CreationAttribute"
  name="cn"
  syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeMap>
  <AttributeDescription
    parent.attr="SunAttribute"
    name="cn"
    syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
  <AttributeDescription
    parent.attr="WindowsAttribute"
    name="cn"
    syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
</AttributeMap>

<AttributeMap>
  <AttributeDescription
    parent.attr="SunAttribute"
    name="uniquemember"
    syntax="1.3.6.1.4.1.1466.115.121.1.25"/>
  <AttributeDescription
    parent.attr="WindowsAttribute"

```

EXAMPLE 7-1 Sample Export Configuration File (Continued)

```

        name="member"
        syntax="1.2.840.113556.1.4.910"/>
</AttributeMap>

<AttributeDescription
    parent.attr="SignificantAttribute"
    name="member"
    syntax="1.2.840.113556.1.4.910"/>
</ActiveDirectoryGlobals>

<SunDirectoryGlobals
    userObjectClass="inetOrgPerson"
    flowInboundCreates="true"
    flowInboundModifies="true"
    flowOutboundCreates="true"
    flowOutboundModifies="true">
<AttributeDescription
    parent.attr="SignificantAttribute"
    name="uniquemember"
    syntax="1.3.6.1.4.1.1466.115.121.1.25"/>
<AttributeDescription
    parent.attr="CreationAttribute"
    name="cn"
    syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeDescription
    parent.attr="SignificantAttribute"
    name="cn"
    syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeDescription
    parent.attr="SignificantAttribute"
    name="pwdaccountlockedtime"
    syntax="1.3.6.1.4.1.1466.115.121.1.24"/>
<TopologyHost
    parent.attr="SchemaLocation"
    hostname="ds-host.example.com"
    port="389"
    portSSLOption="false"
    securePort="636">
    <Credentials
        parent.attr="Credentials"
        userName="cn=directory manager"
        cleartextPassword=""/>
        <!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD -->
    </TopologyHost>
<AttributeDescription
    parent.attr="SignificantAttribute"

```

EXAMPLE 7-1 Sample Export Configuration File (Continued)

```
        name="uid"
        syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeDescription
    parent.attr="CreationAttribute"
    name="sn"
    syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeDescription
    parent.attr="SignificantAttribute"
    name="sn"
    syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
</SunDirectoryGlobals>
</ActiveConfiguration>
```

After the completion of configuration export, `export11cnf` reports the result of the operation. If the operation fails, an appropriate error message is displayed with an error identifier.

Checking for Undelivered Messages

The migration process minimizes system downtime by preserving the connectors' states in the existing deployment. However, these states reflect only the last change received and acknowledged by the Message Queue. Therefore, you do not know whether the message was actually delivered and applied to the destination connector.

This behavior does not cause problems as long as the Message Queue remains the same. However, you will lose any messages on the Message Queue during the migration process when you install Message Queue 3.6.

You must verify that the synchronization topics on the existing Message Queue do not have any undelivered messages before you proceed with the migration. The Identity Synchronization for Windows `checktopics` utility enables you to verify that all the synchronization topics are empty and the system is not causing any problem.

▼ Using the `checktopics` Utility

The `checktopics` utility is delivered in the `migration` directory of the Solaris/SPARC and the Windows Identity Synchronization for Windows 6.0 package.

Note – The prerequisite to run `checktopics` is a Java Virtual Machine.

When you run the `checktopics` utility, it connects to the configuration directory, which contains information about Synchronization User Lists (SULs) and current synchronization

topic names used in Message Queue. In addition, when you run `checktopics`, it queries Message Queue to check how many outstanding messages remain on each active synchronization topic and then displays this information for you.

To execute the `checktopics` command line utility:

- 1 **Open a Terminal window and `cd` to the migration directory.**
- 2 **From a command prompt, type the subcommand as follows.**

```
java -jar checktopics.jar -h hostname \
  -p port -D bind-DN \
  -w bind-password -s root-suffix \
  -q configuration-password -Z
```

For example,

```
java -jar checktopics.jar -D "cn=directory manager" -w - -s "dc=example,dc=com"
-q -Z
```

Note – For more information about the `checktopics` arguments, see “Common Arguments to the Idsync Subcommands” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*. For more information about using `checktopics`, see [“Checking for Undelivered Messages” on page 116](#).

After running `checktopics`, check your terminal for the following messages:

- If the operation succeeds, the terminal window displays a message stating that there are no outstanding messages in the logs.
 - If the operation fails, an appropriate error message is displayed with an error identifier.
-

▼ To Clear Messages

If any of the active synchronization topics contain outstanding messages, use the following procedure to clear the messages.

- 1 **Restart synchronization.**
- 2 **Wait until the messages are applied to the destination connector.**
- 3 **Stop synchronization.**
- 4 **Rerun `checktopics`.**

Forcing Password Changes on Windows NT

On Windows NT, password changes are not monitored and new password values are not captured during the migration process. Consequently, you cannot determine new password values after the migration process.

Instead of requiring all users to change passwords when you finish migrating to 6.0, you can use the `forcepwchg` command-line utility to require a password change for all the users who changed passwords during the migration process.

Note – The `forcepwchg` utility is available only in the Windows packages.

You can find the `forcepwchg` utility in the Windows migration directory. Execute `forcepwchg` directly from that directory. No additional installation steps are necessary.

You must run `forcepwchg` on the Primary Domain Controller (PDC) host where the NT components (connector, Change Detector DLL, and Password Filter DLL) are installed. You cannot run `forcepwchg` remotely.

The `forcepwchg` utility also prints the account names (one name per line) that it is trying to migrate. If an error occurs during the migration process, look into the next entry to the last printed entry.

Migrating Your System

This section provides instructions for migrating a single-host deployment to version Identity Synchronization for Windows 6.0.

In a single-host deployment, all Identity Synchronization for Windows components are installed on a single host (Windows 2000 Server, Solaris version 8 or 9, or SPARC), as follows:

- Directory Server (one instance)
- Core (Message Queue, Central Logger, System Manager, and Console)
- Active Directory Connector
- Directory Server Connector
- Directory Server Plug-in

Note – If you are using Solaris as your installation host, then a Windows 2000 machine with Active Directory is required for synchronization purposes only. (No components would be installed on the Windows 2000 machine.)

The following figure illustrates the migration process and serves as a checklist to supplement the migration instructions that follow.

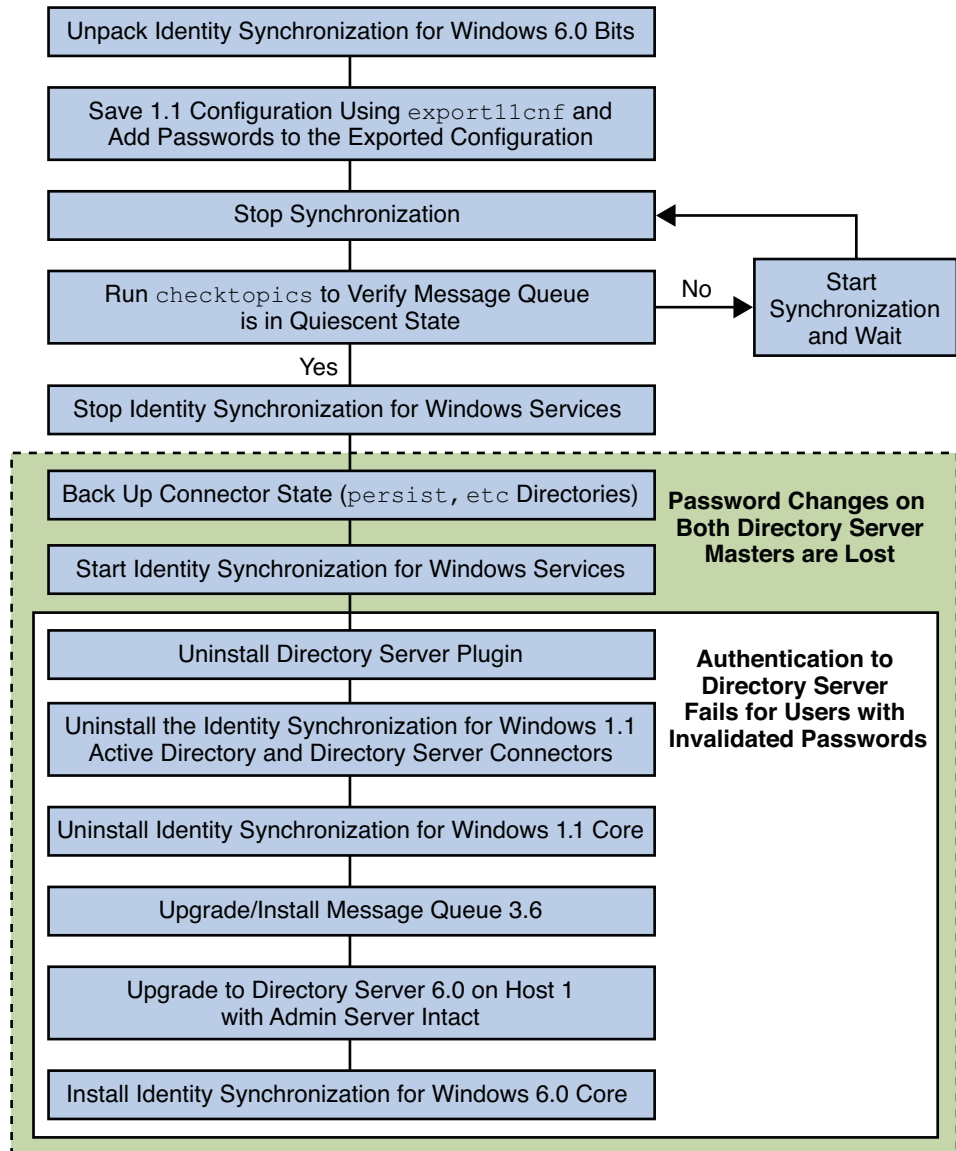


FIGURE 7-1 Migrating a Single-Host Deployment

Preparing for Migration

Use the following procedure to prepare for migration to version 6.0.

▼ **Preparing to migrate from version 1.1, and 1.1 SP1, to version 6.0**

1 Open a terminal window or command prompt.

- **On Solaris** type the following command.

```
uncompress -c filename | tar xf -
```

- **On Windows** type the following command or use any archive program for Windows, such as WinZip.

```
%JAVA_HOME%\bin\jar -xf filename
```

When the binaries are unpacked, the following subdirectories contain the required migration tools:

- installer/
- lib/
- migration/

Solaris	Windows
export11cnf.jar	export11cnf.jar
	forcepwchg.exe
checktopics.jar	checktopics.jar

2 Export your version 1.1 configuration settings to an XML file.

From the migration directory, execute export11cnf as described in [“Using the export11cnf Utility” on page 110](#).

```
java -jar export11cnf.jar -D "cn=directory manager" -w - \
-s "dc=example,dc=com" -q - -f export.cfg
```

3 Add passwords to the exported XML file.

Enter a password between the double quotes for each cleartextPassword field in the exported configuration file. For more information, see [“Inserting Clear-Text Passwords” on page 110](#).

4 Stop synchronization as described in “Starting and Stopping Synchronization” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.

5 Verify that your system is in a stable state.

From the migration directory, execute `checktopics` as described in [“Using the checktopics Utility” on page 116](#). The following example shows the execution of the `checktopics` command.

```
java -jar checktopics.jar -D "cn=directory manager" -w - \
-s "dc=example,dc=com" -q -Z
```

6 Stop Identity Synchronization for Windows services (daemons) as described in “Starting and Stopping Services” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.

Note – Do not stop the Sun ONE Message Queue service.

7 On Windows NT only, perform the following steps.

a. Stop the Sun One NT Change Detector Service by typing the following command.

```
net stop "Sun One NT ChangeDetector Service"
```

b. Save the NT Change Detector Service counters.

i. Open the Registry Editor by executing `regedt32.exe`.

ii. Select the `HKEY_LOCAL_MACHINE` window.

iii. Navigate to the `SOFTWARE\Sun Microsystems\PSW\1.1` node.

iv. Save the following registry values.

- `HighestChangeNumber`
- `LastProcessedSecLogRecordNumber`
- `LastProcessedSecLogTimeStamp`
- `QueueSize`

8 Save the connector states by backing up the `persist` and `etc` directories from the existing 1.1 installation tree.

■ On Solaris, type the following command.

```
cd serverRoot/isw-hostname
tar cf /var/tmp/connector-state.tar persist etc
```

■ On Windows, type the following command.

```
cd serverRoot\isw-hostname
zip -r C:\WINNT\Temp\connector-state.zip persist
etc%JAVA_HOME%\bin\jar -cfM %TEMP%\connector-state.jar persist etc
```

Alternatively, use any archive program for Windows, such as WinZip.

- 9 **Start the Identity Synchronization for Windows services. For more information, see “Starting and Stopping Services” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.**

Uninstalling Identity Synchronization for Windows

Note – The Identity Synchronization for Windows 1.1 uninstall program removes the SUNWjss package if it is not registered for use by another application. In particular, this situation may occur on a Solaris machine if you installed a zip version of Directory Server 5.2, where the uninstall program removes the jss3.jar file from /usr/share/lib/mps/secv1.

If you encounter this situation as you migrate to Identity Synchronization for Windows 6.0, the installer reports that a required file is missing, and logs the file name to the installation log. When this happens, you must reinstall the required patches and restart the installation process. For a list of required patches, see (see “Software Dependency Requirements” in *Sun Java System Directory Server Enterprise Edition 6.2 Release Notes*.

▼ To Uninstall Identity Synchronization for Windows Version 1.1

- 1 **Uninstall the Directory Server plug-in manually and restart each Directory Server where the plug-in was installed.**

Execute the following steps on each Directory Server where the plug-in was installed:

- a. **Remove the following entries from the Directory Server:**

cn=config,cn=pswsync,cn=plugins,cn=configcn=pswsync,cn=plugins,cn=config

For example:

```
ldapdelete -D "cn=directory manager" -w - -p <port \> -c cn=config,
cn=pswsync,cn=plugins,cn=configcn=pswsync,cn=plugins,cn=config
```

- b. **Restart the Directory Server.**

- **On Solaris:** Type `<serverRoot \>/slapd-<hostname \>/restart-slapd`
- **On Windows:** Type `<serverRoot \>\\slapd-<hostname \>\\restart-slapd.bat`

- c. **Remove the Plug-in binaries from the system.**

- **On Solaris:** Type `rm <serverRoot \>/lib/psw-plugin.sorm <serverRoot \>/lib/64/psw-plugin.so`
- **On Windows:** Type `del <serverRoot \>\\lib\\psw-plugin.dll`

- 2 **Change directory (cd) to `<ServerRoot \>\isw-<hostname>` and then use the Identity Synchronization for Windows 1.1 (or 1.1 SP1) uninstallation program to uninstall the version 1.1, and 1.1 SP1, Connectors and Core components.**

Note – You must uninstall Connectors before uninstalling Core components.

- **On Solaris or SPARC:** Type `./runUninstaller.sh`
- **On Windows:** Type `\\runUninstaller.bat`

- 3 **Back up the product registry file and remove Identity Synchronization for Windows related entries from the file.**

The location of the file is as follows:

- **On Solaris:** `/var/sadm/install/productregistry`
- **On Windows:** `C:\\WINNT\\System32\\productregistry`

To remove the Identity Synchronization for Windows related entries from the product registry file, follow the instructions provided in [“Manually Uninstalling 1.1 Core and Instances from Solaris” on page 127](#).

- 4 ***On Windows only.* After uninstalling Core, restart your machine.**

Note – If the uninstall fails, you might have to manually uninstall the Identity Synchronization for Windows components. Instructions are provided in [“What to Do if the 1.1 Uninstallation Fails” on page 127](#)

- 5 ***On Windows only.* Verify that Identity Synchronization for Windows is not running. If necessary, you can stop the service from the command line by typing the following command.**

`net stop “Sun ONE Identity Synchronization for Windows”`

If this service continues running after uninstallation, it causes a sharing violation that prevents you from deleting the instance directory.

- 6 **Remove the Identity Synchronization for Windows instance directory (`isw-<hostname>`).**

Installing or Upgrading the Dependent Products

Use the following steps to upgrade the Java Run Environment, install Message Queue, and upgrade Directory Server.

1. Upgrade the Java 2 Runtime Environment (or Java 2 SDK) on each host (except on Windows NT) where Identity Synchronization for Windows components are installed. The minimum required version is 1.5.0.
 - **Java 2 SDK:** <http://java.sun.com/j2se/1.5.0/install.html>
(<http://java.sun.com/j2se/1.4.2/install.html>)
 - **Java 2 Runtime Environment:** <http://java.sun.com/j2se/1.5.0/jre/install.html>
(<http://java.sun.com/j2se/1.4.2/jre/install.html>)
2. Install Message Queue 3.6 by using the instructions provided in *Sun Java System Message Queue 3.6 Installation Guide*.
3. Upgrade Directory Server to version 6.2. For more information, see [Chapter 1, “Overview of the Migration Process for Directory Server.”](#)

Note – To keep the Administration Server intact, use the `-N` option while migrating Directory Server (configuration and data) to version 6.2. For more information on migrating configuration data and user data, see [“Using `dsmsg` to Migrate Configuration Data” on page 30](#) and [“Using `dsmsg` to Migrate User Data” on page 33](#) respectively.

The Directory Server upgrade preserves your current Directory Server configuration and database.

Installing Identity Synchronization for Windows 6.0

Use the following steps to install the Identity Synchronization for Windows 6.0 components.

▼ To install the Identity Synchronization for Windows 6.0 components:

- 1 **Install Identity Synchronization for Windows Core.** For more information, see “Installing Core” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.
- 2 **Execute `idsync preps` against Directory Server to update the schema.**
 - **On Solaris** type the following commands.

```
cd /opt/SUNWisw/bin
idsync preps arguments\
```
 - **On Windows** type the following commands.

```
cd serverRoot\isw-hostname\bin
idsync preps arguments\
```

For more information about `idsync preps`, see Appendix A, “Using the Identity Synchronization for Windows Command Line Utilities,” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.

3 Import your version 1.1, and 1.1 SP1, configuration XML file by typing the following command.

```
idsync importcnf arguments\
```

Note – If the program detects errors in your input configuration file, an error results. Identity Synchronization for Windows aborts the `importcnf` process and provides the necessary information to correct errors.

For more information about using `idsync importcnf`, see “Using `importcnf`” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.

- 4 Install the Identity Synchronization for Windows 6.0 Connectors. For more information, see “Installing Connectors” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.**
- 5 If you did not select the Configure Identity Synchronization for Windows 6.0 Directory Server Plug-in option while installing Directory Server connector, configure it now. For more information, see Appendix A, “Using the Identity Synchronization for Windows Command Line Utilities,” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.**
- 6 Stop Identity Synchronization for Windows services (daemons) as described in “Starting and Stopping Services” in *Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.**
- 7 On Windows NT only, complete the following steps.**
 - a. Stop the NT Change Detector service by typing the following command.**

```
net stop "Sun Java(TM) System NT Change Detector"
```
 - b. Restore the NT Change Detector Service counters.**
 - i. Open the Registry Editor by executing `regedt32.exe`.**
 - ii. Select the `HKEY_LOCAL_MACHINE` window.**
 - iii. Navigate to the `SOFTWARE\Sun Microsystems\Sun Java(TM) System Identity Synchronization for Windows\1.1` node.**

iv. Double-click on each of the following entries to restore their values (which you saved prior to uninstalling version 1.1).

- HighestChangeNumber
- LastProcessedSecLogRecordNumber
- LastProcessedSecLogTimeStamp
- QueueSize

c. Start the NT Change Detector service by typing the following command.

```
net start "Sun Java(TM) System NT Change Detector"
```

8 Remove the version 6.0 `persist` and `etc` directories (and all their contents) from the instance directory and restore the version 1.1, and 1.1 SP1, `persist` and `etc` directories you backed up in ["Preparing for Migration" on page 119](#).

■ On Solaris, type the following command.

```
cd /var/opt/SUNWisdw
rm -rf etc persisttar xf /var/tmp/connector-state.tar
```

■ On Windows, type the following command.

```
cd serverRoot\isdw-hostname
rd /s etc persist%JAVA_HOME%\bin\jar -xf %TEMP%\ connector-state.jar
```

Alternatively, use any archive program for Windows, such as WinZip.

9 Start the service and the synchronization.

a. Start the Identity Synchronization for Windows service as described in *"Starting and Stopping Services" in Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.

b. Start synchronization as described in *"Starting and Stopping Synchronization" in Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide*.

10 Check the central audit log to verify that there are no warning messages.

Note – If you have customized the version 1.1 log settings, you must manually apply those customizations to your version Identity Synchronization for Windows 6.0 installation. Use the Identity Synchronization for Windows Console to configure your log settings.

What to Do if the 1.1 Uninstallation Fails

If the version 6.0 installation program finds remnants of the version 1.1 system, the installation will fail. Verify that all of the 1.1 components are completely removed from the system before starting the new installation.

If the uninstallation program does not uninstall all of the version 1.1 components, you must manually clean up the Identity Synchronization for Windows product registry and Solaris packages.

Detailed instructions for uninstalling Identity Synchronization for Windows version 1.1 manually are provided in the following sections:

- [“Manually Uninstalling 1.1 Core and Instances from Solaris” on page 127](#)
- [“Manually Uninstalling 1.1 Core and Instances from Windows 2000” on page 132](#)
- [“Manually Uninstalling a 1.1 Instance from Windows NT” on page 137](#)

Note – The instructions provided in this section are for uninstalling Identity Synchronization for Windows *version 1.1, and 1.1 SP1*, only.

Do not use the manual uninstallation procedures provided in the following sections unless the Identity Synchronization for Windows uninstallation program fails.

Manually Uninstalling 1.1 Core and Instances from Solaris

Use the instructions provided in this section to manually uninstall Core from a Solaris machine.

Note – In this section, Identity Synchronization for Windows locations are described in the following manner:

`<serverRoot \>/ isw-<hostname \>`

where `<serverRoot \>` represents the parent directory of the Identity Synchronization for Windows installation location.

For example, if you installed Identity Synchronization for Windows in `/var/Sun/mps/isw-<example\>`, the `<serverRoot\>` would be `/var/Sun/mps`.

▼ To Manually Uninstall Core From a Solaris Machine:

- 1 **Stop all Identity Synchronization for Windows Java processes by typing `/etc/init.d/isw stop` into a terminal window.**

If the preceding command does not stop all of the Java processes, type the following commands.

```
/usr/ucb/ps -gauxwww | grep java  
kill -s SIGTERM process IDs from preceding command
```

- 2 **Stop Message Queue.**

- a. **Type the following command to stop the Message Queue broker.**

```
/etc/init.d/imq stop
```

- b. **Type the following commands to stop any remaining `imq` processes.**

```
* ps -ef | grep imqbroker  
* kill -s SIGTERM process IDs from preceding command
```

- c. **Use one of the following methods to uninstall the broker packages and directories.**

- Use the Message Queue broker uninstall script to uninstall the broker. This script is located in the Identity Synchronization for Windows instance directory on the host where you installed Core.

```
serverRoot/isw-hostname/imq_uninstall
```

- Manually uninstall the packages and directories.
Use the `pkgrm` command to remove the following packages.

```
SUNWac1g  
SUNWi1um  
SUNWi1jx  
SUNWi1len  
SUNWxsrt  
SUNWi1qu  
SUNWjaf  
SUNWi1qfs  
SUNWjhrt  
SUNWi1qdoc  
SUNWi1quc  
SUNWi1qsup  
SUNWi1qr  
SUNWjmail
```

Use the `rm -rf` command to remove the following directories.


```
/etc/imq
/var/imq
/usr/bin/imq*
```

- 3 To remove the Identity Synchronization for Windows 1.1 Solaris packages, run `pkgrm package-name` for each of the packages listed in [“Manually Uninstalling 1.1 Core and Instances from Solaris” on page 127](#).

The following example shows the use of `pkgrm` to uninstall packages.

```
pkgrm SUNWidscm SUNWidscn SUNWidscr SUNWidsct SUNWidsoc
```

Package Name	Description
SUNWidscm	Sun ONE Directory Server Identity Synchronization package for Core components and Connectors.
SUNWidscn	Sun ONE Directory Server Identity Synchronization package for Console help files.
SUNWidscr	Sun ONE Directory Server Identity Synchronization package for Core Components.
SUNWidsct	Sun ONE Directory Server Identity Synchronization package for Connectors.
SUNWidsoc	Sun ONE Directory Server Identity Synchronization package for Object Cache.

Type the following command to verify that all of the packages were removed.

```
pkginfo | grep -i "Identity Synchronization"
```

Note – Run the `pkgrm package-name` command again to check if there are still existing packages due to dependencies.

- 4 Remove the Directory Server Plug-in.
- a. Open the Directory Server Console and select the Configuration tab.
 - b. In the left pane, expand the Plugins node and select the `pswsync` node.
 - c. In the right pane, clear the Enable plug-in check box.
 - d. Click Save.

- e. **From the Directory Server Console, locate and remove the following entry from the Configuration Directory:**
`cn=pswsync,cn=plugins,cn=config`
- f. **Stop Directory Server.**
- g. **Remove the Plugin binary by typing the following command.**
`rm -f serverRoot/lib/psw-plugin.so`
- h. **Restart Directory Server.**
- 5 **Back-up (copy and rename) the current productregistry file located in**
`/var/sadm/install/productregistry.`
- 6 **Manually edit the productregistry file in /var/sadm/install/ to remove the following entries, if present:**

Note –

- For best results, use an XML editor. Alternatively, you can use a standard text editor.
- Some of the following components may not be included in your file.
- You must delete the beginning tag (`<compid\>`), ending tag (`</compid\>`), and all contents in-between both tags). Ellipses are used in the following list to represent any additional text, or tags that are included as part of these tags. See the example on [“Manually Uninstalling 1.1 Core and Instances from Solaris” on page 127.](#)

-
- `<compid\>Identity Synchronization for Windows . . . </compid\>`
 - `<compid\>Core . . . </compid\>`
 - `<compid\>unistaller . . . </compid\>`
 - `<compid\>wpsyncwatchdog . . . </compid\>`
 - `<compid\>setenv . . . </compid\>`
 - `<compid\>Create DIT . . . </compid\>`
 - `<compid\>Extend Schema . . . </compid\>`
 - `<compid\>resources . . . </compid\>`
 - `<compid\>CoreComponents . . . </compid\>`
 - `<compid\>Connector . . . </compid\>`
 - `<compid\>DSConnector . . . </compid\>`
 - `<compid\>Directory Server Plugin . . . </compid\>`
 - `<compid\>DSSubcomponents . . . </compid\>`
 - `<compid\>ObjectCache . . . </compid\>`
 - `<compid\>ObjectCacheDLLs . . . </compid\>`
 - `<compid\>SUNWidscr . . . </compid\>`
 - `<compid\>SUNWidscm . . . </compid\>`
 - `<compid\>SUNWidsct . . . </compid\>`

- `<compid\>SUNWidsn . . . </compid\>`
- `<compid\>SUNWidsoc . . . </compid\>`
- `<compid\>ADConnector . . . </compid\>`

The following is an example `<compid\>` tag. Remove `<compid\>`, `</compid\>`, and all the text and tags in-between.

```
<compid\>Identity Synchronization for Windows
  <compversion\>1.1
    <uniquename\>Identity Synchronization for Windows</uniquename\>
    <compinstance\>1
      <children\>
        <compref\>ADConnector
          <instance\>1
            <version\>1.1</version\>
          </instance\>
        </compref\>
        <compref\>DSSubcomponents
          . . .
        </compref\>
      </children\>
    </compinstance\>
  </compversion\>
</compid\>
```

7 Remove the following Identity Synchronization for Windows directories and files.

a. From the installation location, type the following command.

```
rm -rf serverRoot/isw-hostname
```

b. To remove the bootstrap files, type the following command.

```
rm -rf /etc/init.d/isw
```

8 Clean up the configuration directory as follows:

a. Run the following `ldapsearch` command against the configuration directory where Identity Synchronization for Windows Core is installed to locate the Identity Synchronization for Windows Console subtree:

```
ldapsearch -D "cn=directory manager" -w < password \> -b o=netscaperoot
"(nsnickname=isw)" dn
```

Note – `ldapsearch` is located in Directory Server's `<serverRoot\>/shared/bin/ldapsearch`. For example, `/var/Sun/mps/shared/bin/ldapsearch`

The resulting entry should be similar to the following. Note that the entry always ends with *o=NetscapeRoot*.

```
"cn=Sun ONE Identity Synchronization for Windows,cn=server group,  
cn=myhost.mydomain.com,ou=mydomain.com,o=NetscapeRoot"
```

- b. Use the Directory Server Console to remove the Identity Synchronization for Windows Console subtree and all of the subtrees below it.**

9 Clean up the Identity Synchronization for Windows configuration registry as follows:

- a. Run the following `ldapsearch` command to locate the Identity Synchronization for Windows configuration registry in Directory Server:**

```
ldapsearch -D "cn=directory manager" -w < password \> -b "dc=my,dc=domain"  
"(&(objectclass=iplanetservice)(ou=IdentitySynchronization))" dn
```

The resulting entry should be similar to the following:

```
"ou=IdentitySynchronization,ou=Services,dc=my,dc=domain"
```

- b. Use the Directory Server Console to remove the Identity Synchronization for Windows configuration registry and all of the subtrees below it.**

10 Clean up all other Console-related files as follows:

- a. Remove all the Console jar files by typing:**

```
rm -rf < serverRoot \>/java/jars/isw* For example, /var/Sun/mps/java/jars/isw*
```

- b. Remove all the Console servlet jar files by typing:**

```
rm -rf <serverRoot \>/bin/isw/ For example, /var/Sun/mps/bin/isw/
```

Manually Uninstalling 1.1 Core and Instances from Windows 2000

Use the instructions provided in this section to manually uninstall Core from a Windows 2000 machine.

Note – In this section, Identity Synchronization for Windows locations are described in the following manner:

serverRoot\isw-hostname

where *serverRoot* represents the parent directory of the Identity Synchronization for Windows installation location.

For example, if you installed Identity Synchronization for Windows in C:\Program Files\Sun\mps\isw-example, the *serverRoot* would be C:\Program Files\Sun\mps.

▼ To uninstall Core from a Windows 2000 machine:

- 1 Stop all Identity Synchronization for Windows Java processes using one of the following methods:
 - Select Start → Settings → Control Panel → Administrative Tools → Services to open the Services window. In the right pane, right-click on Identity Synchronization for Windows and select Stop.
 - Open a Command Prompt window and type the following command.
`net stop "Sun ONE Identity Synchronization for Windows"`
 - If the preceding methods do not work, use the following steps to stop the Java processes manually.
 - a. Open the Services window, right-click on Identity Synchronization for Windows, and select Properties.
 - b. From the General tab in the Properties window, select Manual from the Startup type drop-down list.

Note – Although you can view Java processes (such as `pswatchdog.exe`) from the Windows Task Manager, you cannot determine which processes are specifically related to Identity Synchronization for Windows. For this reason, do not stop processes from the Windows Task Manager.

- 2 For a Core uninstallation only, stop the Message Queue using one of the following methods:
 - In the Services window, right-click on iMQ Broker in the right pane and select Stop.

- **From a Command Prompt, type the following command.**
`net stop "iMQ Broker"`
- **If the preceding methods do not work, use the following steps to stop Message Queue manually.**
 - a. **Open the Services window, right-click on iMQ Broker and select Properties.**
 - b. **From the General tab in the Properties window, select Manual from the Startup type drop-down list.**
 - c. **Open the Directory Server Console and select the Configuration tab.**
 - d. **In the left pane, expand the Plugins node and select the pswsync node.**
 - e. **In the right pane, uncheck the Enable plug-in check box.**
 - f. **Click Save.**
 - g. **From the Console, locate and remove the following entry from the Configuration Directory:**
`cn=pswsync,cn=plugins,cn=config`
 - h. **Stop Directory Server.**
You can stop the server using one of the following methods:
 - **In the Services window, right-click on Sun ONE Directory Server 5.2 in the right pane and select Stop.**
 - **Open a Command Prompt window and type the following command.**
`net stop slapd-myhostname`
 - i. **Open Windows Explorer to locate and remove the Plugin binary:**
`<ServerRoot>\>\\lib\\psw-plugin.so`
 - j. **Restart Directory Server.**

- 3 **Open a Command Prompt window and type regedit to open the Registry Editor window.**



Caution – Back up your current registry file before proceeding to [“Manually Uninstalling 1.1 Core and Instances from Windows 2000” on page 132.](#)

- a. **In the Registry Editor, select My Computer in the left pane.**

- b. **Select Registry → Export Registry File from the menu bar.**
 - c. **When the Export Registry File dialog box is displayed, specify a name for the file and select a location to save the backup registry.**
- 4 In the Registry Editor, select Edit → Delete from the menu bar.**
- Remove the following Identity Synchronization for Windows keys from the Windows Registry:
- All entries under
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Identity Synchronization for Windows.
 - All CurrentControlSet and ControlSet (such as ControlSet001, ControlSet002, and so forth) entries under HKEY_LOCAL_MACHINE\SYSTEM*, which includes the following entries (if they exist).
 - ...Control\Session Manager\Environment\< *isw-installation directory* >
 - ...Services\Eventlog\Application\Sun ONE Identity Synchronization for Windows
 - ...Services\Sun ONE Identity Synchronization for Windows
 - ...Services\iMQBroker
- 5 Backup (copy and rename) the current product registry file located in C:\WINNT\system32.**
- 6 Edit the C:\WINNT\system32\productregistry file to remove the following tags:**

Note –

- For best results, use an XML editor. Alternatively, you can use a standard text editor.
- Some of the following components may not be included in your file.
- You must delete the beginning tag (<compid>), ending tag (</compid>), and all contents in-between both tags). Ellipses are used in the following list to represent any additional text and/or tags that are included as part of these tags. See the example “[Manually Uninstalling 1.1 Core and Instances from Windows 2000](#)” on page 132.

-
- <compid>Identity Synchronization for Windows . . . </compid>
 - <compid>Core . . . </compid>
 - <compid>unistaller . . . </compid>
 - <compid>wpsyncwatchdog . . . </compid>
 - <compid>setenv . . . </compid>
 - <compid>Create DIT . . . </compid>
 - <compid>Extend Schema . . . </compid>
 - <compid>resources . . . </compid>
 - <compid>CoreComponents . . . </compid>
 - <compid>Connector . . . </compid>

- <compid>DSConnector . . . </compid>
- <compid>Directory Server Plugin . . . </compid>
- <compid>DSSubcomponents . . . </compid>
- <compid>ObjectCache . . . </compid>
- <compid>ObjectCacheDLLs . . . </compid>
- <compid>ADConnector . . . </compid>

The following is a <compid> tag sample. Remove <compid>, </compid>, and all the text and tags in-between.

```
<compid>Identity Synchronization for Windows
  <compversion>1.1
    <uniquename>Identity Synchronization for Windows</uniquename>
    <compinstance>1
      <children>
        <compref>ADConnector
          <instance>1
            <version>1.1</version>
          </instance>
        </compref>
        <compref>DSSubcomponents
          . . .
        </compref>
      </children>
    </compinstance>
  </compversion>
</compid>
```

7 Remove the Identity Synchronization for Windows installation folder located at *serverRoot\isw-hostname*.

For example, C:\Program Files\Sun\mps\isw-example

8 Clean up the configuration directory as follows:

- a. From a Command Prompt window, run the `ldapsearch` command against the configuration directory where Identity Synchronization for Windows Core is installed to locate the Identity Synchronization for Windows Console subtree.**

Note – `ldapsearch` is located in <*serverRoot*>\shared\bin\ldapsearch.

For example, C:\Program Files\Sun\mps\shared\bin\ldapsearch

```
ldapsearch -D "cn=directory manager" -w <password> -b o=netscaperoot
"(nsnickname=isw)" dn
```

The resulting entry should be similar to the following (note that the entry will always end with *o=NetscapeRoot*):


```
"cn=Sun ONE Identity Synchronization for Windows,cn=server group,
cn=myhost.mydomain.com,ou=mydomain.com,o=NetscapeRoot"
```

- b. **Use the Directory Server Console to remove the Identity Synchronization for Windows Console subtree that you found and all subtrees under it.**

- 9 **Clean up the Identity Synchronization for Windows configuration directory (*also know as the configuration registry*) as follows:**

- a. **From a Command Prompt window, run the following `ldapsearch` command to locate the Identity Synchronization for Windows configuration directory in Directory Server:**

```
ldapsearch -D "cn=directory manager" -w <password \> -b "dc=my,dc=domain"
"(&(objectclass=iplanetservice)(ou=IdentitySynchronization))" dn
```

The resulting entry should be similar to the following:

```
"ou=IdentitySynchronization,ou=Services,dc=my,dc=domain"
```

- b. **Use the Directory Server Console to remove the configuration directory subtree that you found, including all subtrees under it.**

- 10 **Clean up all other Console-related files as follows:**

- a. **Remove all Console jar files located in `<serverRoot\>\java\jars\isw*`For example, `C:\Program Files\Sun\mps\java\jars\isw*`**

- b. **Remove all Console servlet jar files located in `<directory-server-install-root\>\bin\isw\`For example, `C:\SunOne\Servers\bin\isw\`**

Next Steps Restart your machine for all changes to take effect.

▼ Manually Uninstalling a 1.1 Instance from Windows NT

Use the instructions provided in this section to manually uninstall an instance from a Windows NT machine.

Note – In this section, Identity Synchronization for Windows locations are described as follows:

`<serverRoot\>\isw-<hostname\>`

where `<serverRoot \>` represents the parent directory of the Identity Synchronization for Windows installation location. For example, if you installed Identity Synchronization for Windows in `C:\Program Files\Sun\mps\isw- example`, the `< serverRoot \>` would be `C:\Program Files\Sun\mps`.

- 1 **Stop all the Identity Synchronization for Windows Java processes (Core and instance installations) using one of the following methods:**
 - **Select Start → Settings → Control Panel → Administrative Tools → Services to open the Services window. In the right pane, right-click on Identity Synchronization for Windows and select Stop.**
 - **Open a Command Prompt window and type the following command:**
`net stop "Sun ONE Identity Synchronization for Windows"`
 - **If the preceding methods do not work, use the following steps to stop the Java processes manually:**
 - a. **Open the Services window, right-click on Identity Synchronization for Windows, and select Properties.**
 - b. **From the General tab in the Properties window, select Manual from the Startup type drop-down list.**
-

Note – Although you can view Java processes (such as `pswatchdog.exe`) from the Windows Task Manager, you cannot determine which processes are specifically related to Identity Synchronization for Windows. For this reason, do not stop processes from the Windows Task Manager.

- 2 **Stop the Change Detector service using one of the following methods:**
 - **In the Services window, right-click on Sun ONE NT Change Detector Service in the right pane and select Stop.**
 - **Open a Command Prompt window and type the following command:**
`net stop "Sun ONE NT Change Detector Service"`

- If the preceding methods do not work, use the following steps to stop the Change Detector Service manually:
 - a. Open the Services window, right-click on Change Detector Service and select Properties.
 - b. From the General tab in the Properties window, select Manual from the Startup type drop-down list.
 - c. Restart your Windows NT computer.
- 3 You must remove Identity Synchronization for Windows registry keys. Open a Command Prompt window and type `regedt32` to open the Registry Editor window.



Caution – *Do not* use `regedit` because the program does not allow you to edit multi-value strings.

Backup your current Windows registry file before proceeding to [“Manually Uninstalling a 1.1 Instance from Windows NT” on page 137](#).

- a. In the Registry Editor, select the top node (My Computer) in the left pane.
 - b. Select Registry → Export Registry File from the menu bar.
 - c. When the Export Registry File dialog box is displayed, specify a name for the file and select a location to save the backup registry.
- 4 In the Registry Editor, select Edit → Delete from the menu bar.

Remove the following Identity Synchronization for Windows keys from the Registry:

- All entries under
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Identity Synchronization for Windows
- All CurrentControlSet and ControlSet (such as ControlSet001, ControlSet002) entries under HKEY_LOCAL_MACHINE\SYSTEM*. These entries include the following:
 - ...\\Control\\Session Manager\\Environment\\ <isw-installation directory>
 - ...\\Services\\Eventlog\\Application\\Sun ONE Identity Synchronization for Windows
 - ...\\Services\\Sun ONE Identity Synchronization for Windows
 - ...\\Services\\iMQBroker
- The HKEY_LOCAL_MACHINE\SOFTWARE\Sun Microsystems\PSW

- 5 Use **regedt32** (*do not use regedit*) to modify (do not delete) the following registry key:
 - a. Select the registry key entry in the left pane:
 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\CONTROL\LSA
 The registry value type must be REG_MULTI_SZ.
 - b. In the right pane, right-click on the Notification Packages value and select Modify.
 - c. Change the PASSFLT value to FPNWCLNT.
- 6 Backup (copy and rename) the current productregistry file located in C:\WINNT\system32.
- 7 Edit the C:\WINNT\system32 productregistry file to remove the following tags:

Note –

- For best results, use an XML editor. Alternatively, you can use a standard text editor.
 - Some of these components might not be included in your file.
 - You must delete the beginning tag (<compid>), ending tag (</compid>), and all contents in-between both tags). Ellipses are used in the following list to represent any additional text and/or tags that are included as part of these tags. See the example on [“Manually Uninstalling 1.1 Core and Instances from Windows 2000”](#) on page 132.
-

- <compid>Identity Synchronization for Windows . . . </compid>
- <compid>Core . . . </compid>
- <compid>uninstaller . . . </compid>
- <compid>wpsyncwatchdog . . . </compid>
- <compid>setenv . . . </compid>
- <compid>Create DIT . . . </compid>
- <compid>Extend Schema . . . </compid>
- <compid>resources . . . </compid>
- <compid>CoreComponents . . . </compid>
- <compid>Connector . . . </compid>
- <compid>DSConnector . . . </compid>
- <compid>Directory Server Plugin . . . </compid>
- <compid>DSSubcomponents . . . </compid>
- <compid>ObjectCache . . . </compid>
- <compid>ObjectCacheDLLs . . . </compid>
- <compid>ADConnector . . . </compid>

The following is a example `<compid\>` tag. Remove `<compid\>`, `</compid\>`, and all the text and tags in-between.

```
<compid\>Identity Synchronization for Windows
  <compversion\>1.1
    <uniquename\>Identity Synchronization for Windows</uniquename\>
      <compinstance\>1
        <children\>
          <compref\>ADConnector
            <instance\>1
              <version\>1.1</version\>
            </instance\>
          </compref\>
          <compref\>DSSubcomponents
            . . .
          </compref\>
        </children\>
      </compinstance\>
    </compversion\>
  </compid\>
```

- 8 **Remove the Identity Synchronization for Windows installation folder located at `<serverRoot\>\isw-<hostname\>`.**

For example, `C:\Program Files\Sun\mps\isw-example`

Note – You must edit the Windows registry as described in [“Manually Uninstalling a 1.1 Instance from Windows NT” on page 137](#) before proceeding to [“Manually Uninstalling a 1.1 Instance from Windows NT” on page 137](#).

- 9 **Remove the Password Filter DLL.**

Locate the `passflt.dll` file in the `C:\winnt\system32` folder, and rename the file to `passflt.dll.old`.

- 10 **Restart your machine for all changes to take effect.**

Other Migration Scenarios

Because other deployment topologies are possible, your migration process may differ from the process described for a single-host deployment.

This section describes two alternative deployment scenarios and explains how to migrate in each case.

The sample deployment scenarios include:

- “Multi-Master Replication Deployment” on page 142
- “Multi-Host Deployment with Windows NT” on page 143

Multi-Master Replication Deployment

In a multi-master replication (MMR) deployment, two Directory Server instances are installed on different hosts. It is possible to run the hosts on different operating systems, but in this scenario, both hosts are running on the same operating system.

Table 7-1 and Figure 7-2 illustrate how the Identity Synchronization for Windows components are distributed between the two hosts.

TABLE 7-1 Component Distribution in a Multi-Master Replication Deployment

Host 1	Host 2
Directory Server (one instance) as the secondary master for synchronized users	Directory Server (one instance) as the preferred master for synchronized users
Core (Message Queue, Central Logger, System Manager, and Console)	Directory Server Plugin
Active Directory Connector	
Directory Server Connector	
Directory Server Plugin	

The migration process keeps on-demand password synchronization running continuously on the preferred master or on the secondary master.

Note – If both hosts are running on a Solaris operating system, then a third host running Windows 2000 with Active Directory is required for synchronization purposes only. (No components would be installed on the third host.)

The following figure illustrates the process for migrating Identity Synchronization for Windows in a MMR deployment.

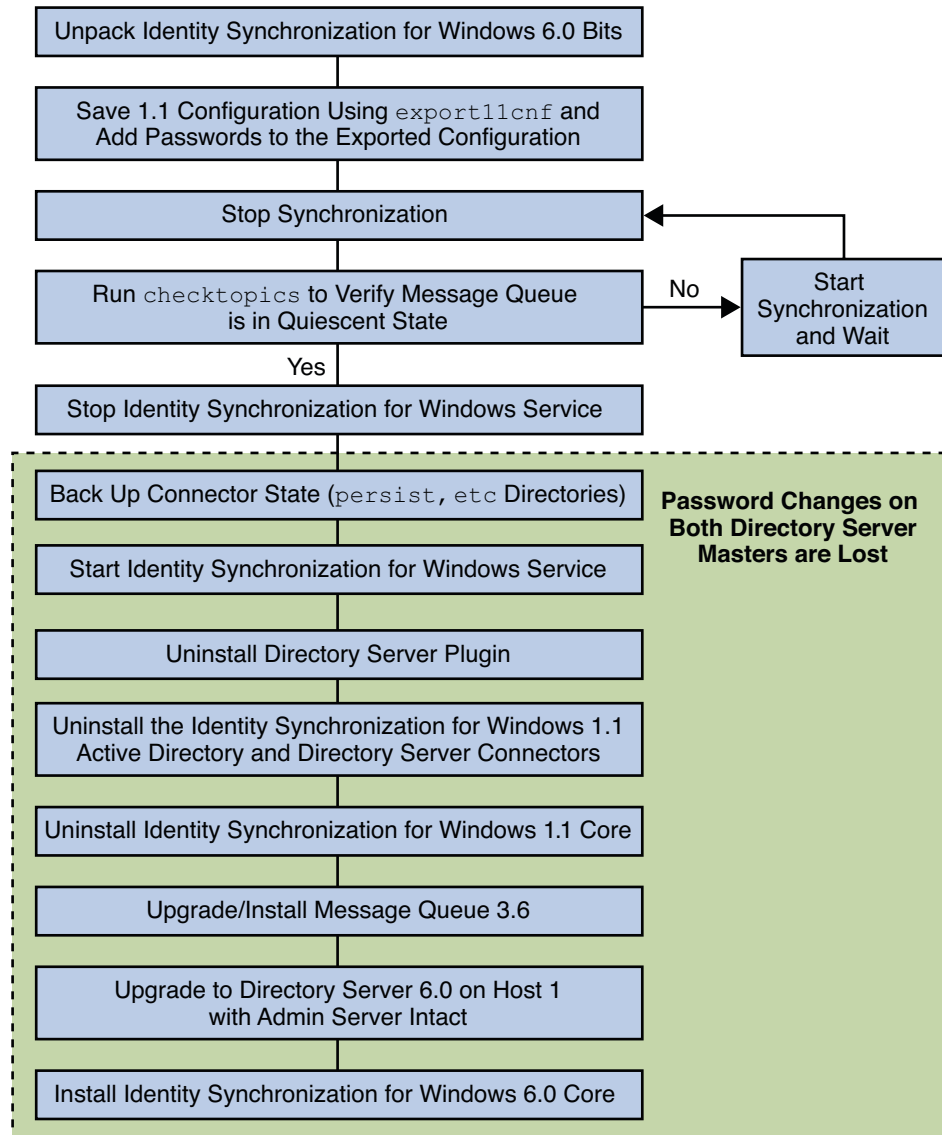


FIGURE 7-2 Migrating a Multi-Master Replication Deployment

Multi-Host Deployment with Windows NT

Three hosts are used in this deployment scenario:

- A Windows NT system
- A host for Directory Server with the synchronized users and the Directory Server Connector

- A host for all other components

Table 7–2 and Figure 7–3 illustrate how the Identity Synchronization for Windows components are distributed between the three hosts.

TABLE 7–2 Multi-Host Deployment

Host 1	Host 2	Host 3
Directory Server with configuration repository	Directory Server for synchronized users	Windows NT Connector
Core (Message Queue, Central Logger, System Manager, and Console)	Directory Server Connector	Windows NT Subcomponents (Password Filter DLL and Change Detector Service)
Active Directory Connector	Directory Server Plugin	

In the previous scenario, hosts 1 and 2 are running on the same operating system.

Note – Directory Server at host1 contains the configuration registry and the Admin Server console. Ensure you migrate to Directory Server 6.0 using the -N option to keep the Admin Server intact. For more information on migrating configuration data and user data, see [“Using dsrmig to Migrate Configuration Data” on page 30](#) and [“Using dsrmig to Migrate User Data” on page 33](#) respectively.

Directory Server at host2 contains the data and the Directory Server plugin. When you migrate Directory Server to 6.0, the plugin configuration is lost. But it does not cause any problem as Identity Synchronization for Windows migration requires the connectors to be reinstalled and plugin to be reconfigured. Therefore, Directory Server at host2 should be migrated after Identity Synchronization for Windows uninstallation.

If both hosts are running a Solaris operating system, then a fourth host running Windows 2000 with Active Directory is required for synchronization purposes only. (No components would be installed on the fourth host.)

Figure 7–3 illustrates the process for migrating Identity Synchronization for Windows for a multi-host deployment

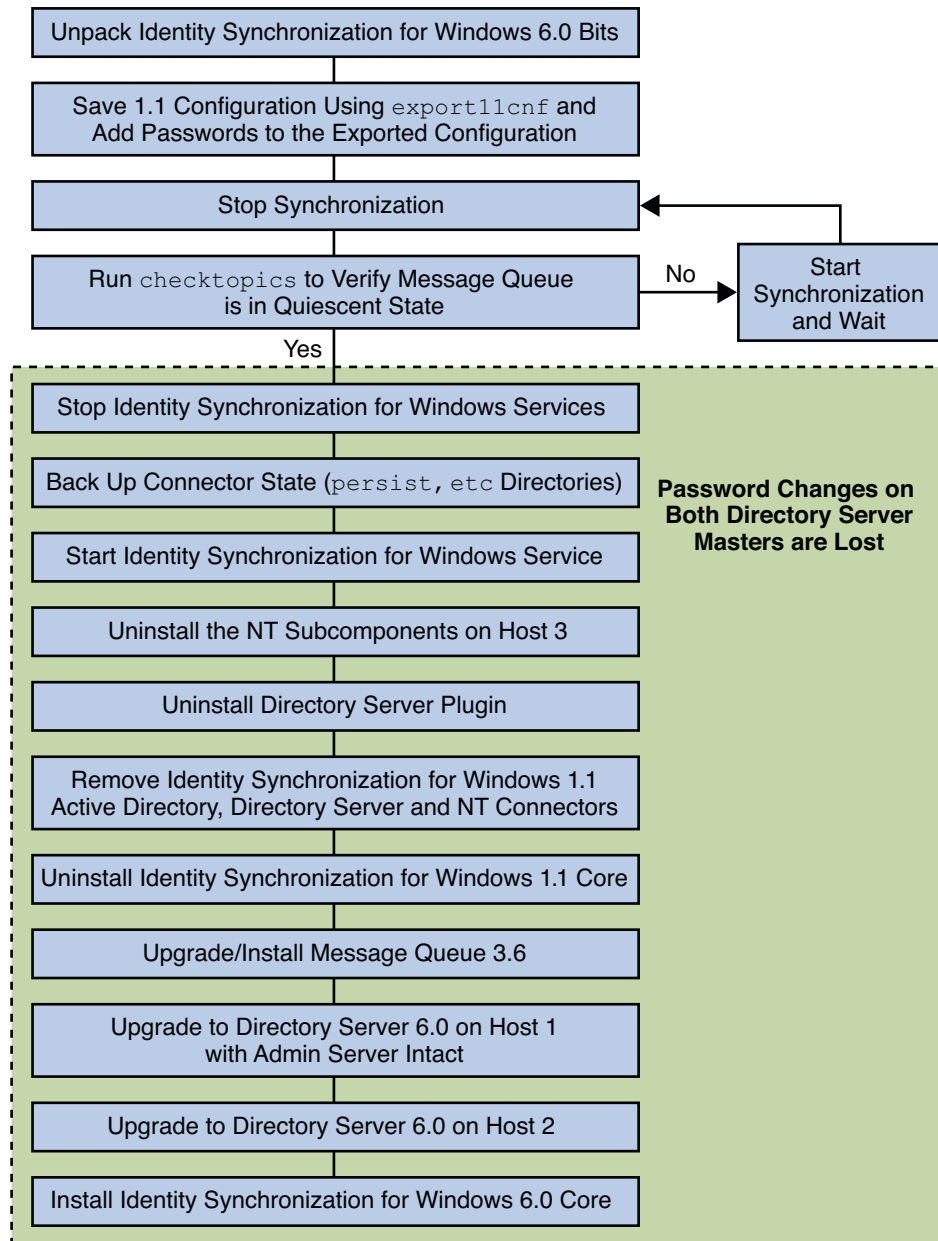


FIGURE 7-3 Migrating a Multi-Host Deployment with Windows NT

Checking the Logs

After migration, check the central audit log for messages indicating a problem. In particular, check for Directory Server users whose password changes may have been missed during the migration process. Such errors would be similar to the following:

```
[16/Apr/2004:14:23:41.029 -0500] WARNING    14  CNN101 ds-connector-host.example.com
      "Unable to obtain password of user cn=JohnSmith,ou=people,dc=example,dc=com,
      because the password was encoded by a previous installation of
      Identity Synchronization for Windows Directory Server Plugin.
      The password of this user cannot be synchronized at this time.
      Update the password of      this user again in the Directory Server."
```

You will not see this log message until after you start synchronization in Identity Synchronization for Windows 6.0. This is why checking the logs is the last step of the migration procedure.

Index

A

- Active Directory
 - during migration, 118
 - hosts, 142, 144
 - MMR deployments, 142
 - multi-host deployments, 144
 - on-demand password synchronization, 108
 - password synchronization during migration, 108
 - synchronizing passwords, 108
- adding, passwords to exported XML files, 120
- arguments
 - checktopics, 117
 - importcnf, 125

B

- binary files
 - removing, 130
 - unpacking, 120

C

- central log directories, 18
- certificate database, default path, 17
- Change Detector subcomponents, 125, 126, 138, 144
- checktopics.jar, 120, 121
- checktopics utility
 - checktopics.jar, 120
 - clearing messages, 117
 - default location, 116

checktopics utility (*Continued*)

- description, 116
- prerequisites, 116
- syntax, 117
- using, 116
- clear-text passwords, inserting, 110-111
- configurations, exporting, 109
- configuring, Identity Synchronization for Windows, 109
- connectors, uninstalling, 123
- consoles
 - help files, 129
 - MMR configuration, 142
 - multi-host deployments, 144
 - removing jar files, 132, 137

Core

- uninstalling, 123, 127, 132
- creating
 - XML configuration documents, 109

D

- default locations, 16-19
- deployments
 - exporting topologies to XML documents, 109
 - MMR, 142
 - multi-host, 143, 144
- deployments, single-host, 118
- detecting, errors, 125
- directories
 - etc, 126

directories (*Continued*)

- isw-hostname, 123, 127, 133
- migration, 109, 110, 116, 118
- persist, 126

Directory Server

- command line changes, 69-71
- restarting, 122
- upgrading, 124

Directory Server Plug-in

- synchronizing password changes, 108
- uninstalling, 122

Directory Server Plugin, removing, 129

E

editing, product registry file, 135

errors

- detecting, 125
- XML configuration file, 125

etc directory

- backing up, 110, 121
- removing, 126
- restoring, 126

examples

- checktopics command, 117
- export11cnf command, 110
- idsync importcnf, 111

export11cnf.jar, 110, 120

export11cnf utility, 109

- description, 109
- export11cnf.jar, 120
- inserting clear-text passwords, 110

exporting

- 1.1 (or 1.1 SP1) configuration, 109, 110
- version 1.1, and 1.1 SP1, configuration files, 109

F

failures, uninstallation, 127

forcepwhchg utility

- description, 109
- forcing password changes, 118
- location, 118

forcepwhchg utility (*Continued*)

- preparing for migration, 120
 - requiring password changes, 118
- forcing password changes, 118

H

help, removing help files, 129

hosts

- Active Directory, 142, 144
- deployment scenarios, 143

I

Identity Synchronization for Windows,

- configuring, 109

idsync importcnf

- examples, 111
- importing configuration files, 109, 125

importcnf subcommand

- examples, 111
- importing configuration files, 109, 125

install-path, 17*instance-path*, 17

instances, uninstalling 1.1 (or 1.1 SP1), 137

isw-hostname directory, 17*isw-hostname* directory, 123, 127, 133**J**

jar files

- checktopics, 120, 121
- export11cnf, 120
- exportcnf, 110
- jss3.jar, 122
- migration tools, 120

Java Naming and Directory Interface, 15

Java processes, stopping, 133

jss3.jar files, removing, 122

L

LDAP, ldapsearch, 131
 ldapsearch, using, 131
 local log directory, 18

M

Message Queue, 16, 133
 upgrading, 124
 migration
 checking for undelivered messages, 116
 clearing messages, 117
 directory, 109, 110, 116, 118
 exporting 1.1 (or 1.1 SP1) configuration, 109
 forcing password changes, 118
 from version 1.1 to 1 2004Q3, 107
 preparing, 119
 scenarios, 141, 142
 using checktopics, 116
 MMR
 deployments, 142
 migration scenarios, 142
 multi-host deployments, 143, 144

P

packages
 removing, 129
 SUNWidscm, 129
 SUNWidscn, 129
 SUNWidscr, 129
 SUNWidsct, 129
 SUNWidsoc, 129
 SUNWjss, 122
 Password Filter subcomponents, 144
 password synchronization, on demand, 108
 passwords
 clear-text, inserting, 110-111
 forcing changes, 118
 synchronizing changes with Directory Server
 Plugin, 108
 PDC, running forcepchg utility, 118

persist directory
 backing up, 110, 121
 restoring, 126
 preparing, for migration, 119
 prerequisites, for checktopics utility, 116
 processes, stopping, 133

R

regedt32.exe, 121, 125, 139, 140
 registries, editing, 135
 removing
 binary files, 130
 console jar files, 132, 137
 Directory Server Plugin, 129
 help files, 129
 packages, 129
 Solaris packages, 129
 restarting
 Directory Server, 122
 synchronization, 117
 restoring, directories, 126

S

schema, updating, 124
 serverroot directory, 17
 single-host, deployments, 118
 SLAMD Distributed Load Generation Engine, 15
 Solaris, removing packages, 129
 stopping
 Java processes, 133
 Message Queue, 133
 subcommands
 importcnf, 109, 111, 125
 SUNWidscm package, 129
 SUNWidscn package, 129
 SUNWidscr package, 129
 SUNWidsct package, 129
 SUNWidsoc package, 129
 SUNWjss package, removing, 122
 synchronization, restarting, 117

- synchronizing, changes with Directory Server
 - Plug-in, 108
- syntax
 - checktopics command, 117
 - checktopics utility, 117
 - export11cnf command, 110
- system, verifying quiescence, 116

- XML configuration documents (*Continued*)
 - exporting configurations, 109, 110

U

- uninstallation failures, 127
- uninstalling
 - 1.1 (or 1.1 SP1) instances, 137
 - connectors, 123
 - Core, 123, 127, 132
 - Directory Server Plug-in, 122
- UNIX commands
 - removing binaries, 122
 - restarting Directory Server, 122
 - uninstalling program, 123
- unpacking product binary files, 120
- updating, schema, 124
- using, checktopics utilities, 116
- utilities
 - checktopics, 109, 116
 - export11cnf, 109
 - forcepwchg, 109
 - using checktopics, 116

V

- verifying
 - empty synchronization topics, 116
 - system quiescence, 116

X

- XML configuration documents
 - creating, 109
 - errors, 125
 - export11cnf, 109, 110